ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

DOTTORATO DI RICERCA IN

COMPUTER SCIENCE & ENGINEERING

Ciclo XXXII

Settore Concorsuale: 09/H1

Settore Disciplinare: ING-INF/05

Hybrid Offline/Online Methods for Optimization Under Uncertainty

Presentata da:
Allegra De Filippo

Coordinatore Dottorato:
Prof. Davide Sangiorgi

Supervisore:
Prof. Michela Milano

Esame Finale Anno 2020

"*Exploring the unknown requires tolerating uncertainty.*"

(Brian Greene)

Abstract

This work considers multi-stage optimization problems under uncertainty. In this context, at each stage some uncertainty is revealed and some decision must be made: the need to account for multiple future developments makes stochastic optimization incredibly challenging. Due to such a complexity, the most popular approaches depend on the temporal granularity of the decisions to be made. These approaches are, in general, sampling-based methods and heuristics. Long-term *strategic* decisions (which are often very impactful) are typically solved via expensive, but more accurate, sampling-based approaches. Short-term *operational* decisions often need to be made over multiple steps, within a short time frame: they are commonly addressed via polynomial-time heuristics, while more advanced sampling-based methods are applicable only if their computational cost is carefully managed. We will refer to the first class of problems (and solution approaches) as *offline* and to the second as *online*. These phases are typically solved in isolation, despite being strongly interconnected. Starting from the idea of providing multiple options to balance the solution quality/time trade-off in optimization problem featuring offline and online phases, we propose different methods that have broad applicability. These methods have been firstly motivated by applications in real-word energy problems that involve distinct offline and online phases: for example, in Distributed Energy Management Systems we may need to define (offline) a daily production schedule for an industrial plant, and then manage (online) its power supply on a hour by hour basis. Then we show that our meth-

ods can be applied to a variety of practical application scenarios in very different domains with both discrete and numeric decision variables.

In the first part of this thesis, we propose general methods based on a tighter integration between the two phases and we show that their applicability can lead to substantial improvements. Our methods are applicable under two (fairly general) conditions: 1) the uncertainty is exogenous; 2) it is possible to define a greedy heuristic for the online phase that can be modeled as a parametric convex optimization problem. We start with a baseline composed by a two-stage offline approach paired with an online greedy heuristic. We then propose multiple methods to tighten the offline/online integration, leading to significant quality improvements, at the cost of an increased computation effort either in the offline or the online phase.

The second part of this thesis focuses on how to manage the cost/quality trade-off of online stochastic anticipatory algorithms, taking advantage of some offline information. Sampling-based anticipatory algorithms can be very effective at solving online optimization problems under uncertainty, but their computational cost may be sometimes prohibitive. In many practical cases, some degree of information about future uncertainty is available significantly in advance. This provides an opportunity to exploit offline techniques to boost the performance of the online method. In this context, we present three methods that, given an arbitrary anticipatory algorithm, allow to retain its solution quality at a fraction of the online computational cost, via a substantial degree of offline preparation. Our approaches are obtained by combining: 1) a simple technique to identify likely future outcomes based on past observations; 2) the (expensive) offline computation of a "contingency table"; and 3) an efficient solution-fixing heuristic.

Overall, all the methods proposed in this thesis provide multiple options to balance the solution quality/time trade-off in optimization problem featuring offline and online phases, suiting a variety of practical application scenarios. We ground our methods on two real case studies

with both offline and online decisions: an energy management system with uncertain renewable generation and demand, and a routing problem with uncertain travel times. The application domain feature respectively continuous and discrete decisions. An extensive analysis of the experimental results shows that indeed offline/online integration may lead to substantial benefits by achieving high solution quality, while reducing the online computation time.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis considers multi-stage optimization problems under uncertainty that involve distinct offline and online phases. In particular it addresses the issue of integrating these phases to show how the two are often interrelated in real-world applications.

1 Context

Optimization under uncertainty arises in many application areas, such as project scheduling, transportation systems, financial systems, and energy management: fuel prices, electrical power, activity durations, travel times, etc. are effectively stochastic in the real world. Optimization problems in this class can be seen as a sequence of multiple stages, such that at each stage part of the uncertainty is revealed and some decisions must be made. Such decisions are irrevocable and made without full knowledge of the future: they should therefore account for multiple (ideally all) possible outcomes, and optimize a probabilistic performance measure (e.g.the expected value of a relevant cost metric). The need to account for multiple future developments makes stochastic optimization incredibly challenging, which explains how approximate (sampling-based) methods and heuristics are the most popular solution techniques. Due to such a complexity, the applicable approaches

depend on the temporal granularity of the decisions to be made. Long-term strategic decisions (which are often very impactful) are typically solved via expensive, but more accurate, sampling-based approaches. Short-term operational decisions often need to be made over multiple steps, within a short time frame: they are commonly addressed via polynomial-time heuristics, while more advanced sampling-based methods are applicable only if their computational cost is carefully managed. We will broadly refer to the first class of problems (and solution approaches) as offline and to the second as online.
In this thesis, we move from the observation that *many practical application scenarios require to make interdependent offline and online decisions*. For example, we may need to define a daily production schedule for an industrial plant, and then manage its power supply on a hour by hour basis; or we may assign customers to vehicles for delivering goods, and then adjust their routes dynamically as the traffic conditions reveal themselves over time. As we can see in the related current literature, the most common approach to tackle such problems is to deal with the offline and online phase separately. Sampling-based online algorithms have become more common and heuristics are still the most common approach in practical applications, though. We will show that *substantial improvements can be obtained by treating the two phases in an integrated fashion*.

2 Contribution

This distinction in offline and online optimization has led to consider the two modes separately in recent literature. However, in many cases multi-stage optimization problems under uncertainty can be considered composed of both an offline strategic phase and an operational online phase. Strict constraints on the available decision time are often present in the online phase, but are absent (or very relaxed) on the offline one. In this thesis, we will show that a tighter integration between the two

phases can lead to substantial improvements: this will be done via an empirical evaluation using the first two examples (energy management and transportation system) as case studies.

In the first part of this thesis, we start from a general baseline model that integrate both offline and online decision phases and then we improve it by altering either the offline or the online component of the solution process, so that the two play better together. In the second part, we start from a generic sampling-based online anticipatory algorithm and we propose methods to show how to exploit the existence of an offline phase to manage its cost/quality trade-off.

In the first part, as a baseline, we consider an approach that deals with offline decisions via a sampling-based method, and with online decisions via a greedy heuristic. This baseline is not problem specific, instead we simply assume that: 1) the uncertainty is exogenous; 2) a two-stage stochastic optimization model is used for the offline phase; 3) the online heuristic can be stated as convex optimization problem. We then show how to improve the baseline in different directions, each *altering either the offline or the online component of the solution process*. The baseline is improved via three broad ideas: 1) improving the online heuristic by adding an anticipatory component; 2) making the offline solver aware of the online heuristic and its limitations; 3) tuning the parameters of the online heuristic to alter its behavior. The first idea is closely related to existing online anticipatory algorithms (e.g. EXPECTATION); the second and third ideas exploit the mixed nature of the problem to enable improvements via a deeper integration of the offline and online phases. We formalize our methods to propose general approaches that can be applied to different real world use cases, as long as a few basic assumptions are satisfied. We believe our techniques represent a significant step toward integrated offline/online optimization. To test our methods, we ground them on two case studies, matching the examples mentioned earlier: 1) an energy system management problem, where load shifts are planned offline (the day ahead) and power flows

must be controlled online (e.g. hour by hour); and 2) a Vehicle Routing Problem where customer are assigned offline, but the routes can be chosen online (i.e. based on the uncertain travel times). The first problem features a continuous (and hence non-enumerable) decision space, while the second has pure discrete decisions. In our experiments, all the proposed methods significantly improve over the baseline in terms of solution quality. While the computation cost is always higher than the baseline, each approach hits a different trade-off in terms of offline and online solution time.

In the second part of the thesis, we focus on *stochastic online anticipatory algorithms* that have a considerable computational cost, which may be problematic if (as it is often the case) online decisions must be taken within a short time frame. In most practical settings, however, a substantial amount of time and information is available before the online problem is solved, in an offline phase. For example, one may have access to energy production forecasts, historical travel times in routing problems, results from test runs in cyber-physical systems. We refer to this sort of data as offline information. Usually, it is employed to characterize the uncertain elements and for sampling likely outcomes (i.e. scenarios). We will show how to exploit this information at a much deeper level. In this context, we propose three *hybrid offline/online methods* that build over a given, sampling-based, anticipatory algorithm, and allow to match its solution quality at a fraction of the online computational cost. One of them can even rely on a deterministic algorithm, thus providing state-of-the-art performance in problems for which no anticipatory approach is available. All our methods work by shifting part of the computation to the offline phase, where time limits are more relaxed and the costs can be better amortized (e.g. via parallelization). We obtain our methods by combining three basic contributions: 1) a technique to estimate the probability of future outcomes, given past observations; 2) a scheme for building a contingency table, with precomputed solutions to guide the online choices; and 3)

an efficient fixing heuristic for adapting the precomputed solutions to run-time conditions. We ground our approaches on a (numeric) energy management problem with uncertain loads and generation from Renewable Energy Sources (RES), and on a (discrete) Traveling Salesman Problem with uncertain travel times. We show how our methods reach a solution quality comparable with the anticipatory algorithm, with lower (or dramatically lower) online computational cost.
Our contributions can be summarized as follows:

1. We propose multiple methods to tighten the offline/online integration, leading to significant quality improvements, at the cost of an increased computation effort either in the offline or the online phase.
2. Given an arbitrary online (sampling-based) anticipatory algorithm, we propose three methods that allow to retain its solution quality at a fraction of the online computational cost, via a substantial degree of offline preparation.

All our methods have broad applicability and they provide multiple options to balance the solution quality/time trade-off in optimization problem featuring offline and online phases, suiting a variety of practical application scenarios in very different domains with both discrete and numeric decision variables.

3 Outline

The structure of the thesis is the following.
Chapter 2 provides a review of methods for optimization under uncertainty, both in an offline and online setting, and then focuses on motivating examples. The chapter also provides an introduction and state-of-the-art discussion on the enabling methodologies used to tackle the problem of optimization under uncertainty in complex systems. In particular, all the methods proposed in this work were originally born for

the energy system domain, which presents integrable offline/online decisions. For this reason, in the final part of this chapter, a section is devoted to the optimization techniques and description of Distributed Energy Systems (e.g. Virtual Power Plants).
Chapter 3 formally describes our integrated offline/online proposed methods (as improvements of a detailed baseline model) by pointing out the importance of both the offline and the online part for each method.
In Chapter 4 we ground the methods developed in Chapter 3 on two real case studies and we provide an exhaustive analysis and discussion of the results.
Chapter 5 formally introduces our three hybrid methods that, given an arbitrary online anticipatory algorithm, allow to retain its solution quality at a fraction of the online computational cost, via offline preparation.
In Chapter 6 we ground the methods developed in Chapter 5 on two real case studies and we provide an exhaustive analysis and discussion of the results.
Chapter 7 concludes with the final remarks and future works.

Chapter 2

Related Work

In this section we provide an overview of methods for optimization under uncertainty, both in an offline and online setting, and then we provide motivating examples to show how both the offline and online phases are often interrelated in real-world applications. Moreover, since the work described in this thesis was originally motivated by problems on the energy domain, the final part of this chapter is dedicated to optimization techniques and description of distributed energy systems such as Virtual Power Plants.

1 Optimization Under Uncertainty

Optimization under uncertainty is characterized by the need to make decisions without complete knowledge about the problem data. This situation is extremely common, and there is a growing realization that dealing with uncertainty in optimization is necessary to achieve real-world impact in many domains.

A large number of problems in production planning and scheduling, transportation, energy management, and finance require that decisions be made in the presence of uncertainty (e.g. electricity and fuel prices, renewable energy production, routing travel times). However, this situation is very challenging: ideally, one should optimize for every possible

contingency, which is often impossible or impractical, and a key difficulty is in dealing with an uncertainty space is that it is huge and frequently leads to very large-scale optimization models. Decision-making under uncertainty is often further complicated by the presence of decision variables in a multi-period or multi-stage setting. Optimization problems in this class can be seen as a sequence of multiple stages, such that at each stage part of the uncertainty is revealed and some decisions must be made. Such decisions are irrevocable and made without full knowledge of the future: they should therefore account for multiple (ideally all) possible outcomes, and optimize a *probabilistic* performance measure (e.g.the expected value of a relevant cost metric).

One extreme (and frequent) method to deal with such issues is to disregard the uncertainty and assume that all parameters are deterministic [Sah04]. When the potential impact of uncertainty is not negligible, however, using stochastic optimization becomes necessary (see [SP07] for an introduction or [BL97, KWK94] for an extensive discussion). In this case, *a suitable representation for the uncertainty must be found* and (except in rare cases) some technique must be used to *trade estimation accuracy for a reduction of the computation time*. The field has been extensively investigated, and we refer the reader to [Pow16] for a comprehensive overview.

There are two main approaches to deal with data uncertainty in optimization, namely robust and stochastic optimization. Robust optimization does not assume that probability distributions of uncertain data are known, but instead it assumes that the uncertain data resides in the so-called uncertainty set. Additionally, basic versions of Robust Optimization assume hard constraints, i.e., constraint violation cannot be allowed for any realization of the data in the uncertainty set. The Robust approach is popular because of its computational tractability for many classes of uncertainty sets and problem types. However, when distributions of the uncertainty are sufficiently well characterized, the Stochastic approach may be better. Stochastic optimization has an im-

portant assumption, i.e., the true probability distribution of uncertain data has to be known or estimated. If this condition is met and the reformulation of the uncertain optimization problem is computationally tractable, then the stochastic approach is arguably the most effective methodology to solve the uncertain optimization problem.
We discuss both the approaches in the following sections.

1.1 Robust Optimization

When a deterministic model is inappropriate, and there are few probability indications for using a stochastic model, it could be useful to work with ranges of uncertainty. Uncertain parameters, in this case, are assumed as restricted to particular intervals, without an associated probability distribution. This is the key idea in Robust Optimization and has the additional benefit of reducing even further the computational costs [BS04, BBC11, ZWL15, Nem, BTN08]. Instead of minimizing the total expected cost as in stochastic optimization, robust optimization reduces the worst-case costs for all possible results of uncertain parameters. Often the objective is to make a trade-off between robustness and the solution quality in the most common scenarios. An uncertainty set that contains all possible realizations for each component of the uncertain parameters is the most robust choice, but on the other hand there is only a small chance that all uncertain parameters take their worst case values. Since often the underlying probability distribution is not known, the idea is to find a distributionally robust solution [GYdH15]. Robust approaches might lead to a substantially higher costs of the proposed solution [BS04] w.r.t. stochastic ones when distributions of the uncertainty are sufficiently well characterized. This is mainly because robust approaches protect against each event in the specified uncertainty set regardless of its probability, and therefore may have to account for extremely unlikely events. Several robust approaches have parameters (e.g., budget of uncertainty) that can be used to adjust the degree of protection offered by the model [CSS07]; yet,

in general tuning these parameters is not trivial. To reduce the price of robustness, subsequent studies have investigated alternative soft and light robustness models [FM09, BTBB10]. Recently, multiband robustness [BD12], has been proposed to support an improved and stratified representation of uncertainty, while maintaining the computational tractability. It would be worth mentioning also Model-predictive control (MPC) algorithms [BM99] that are affinely adjustable robust optimization where the parameters of a state function (that acts online to changes in state/uncertainty) are optimized offline. Such an algorithm nicely spans the space between a myopic heuristic, and a two-stage anticipatory algorithm with many scenarios. Hybrid stochastic/robust models have been proposed in recent years (e.g. [ZG13, KRK16, LXT15]) to combine the advantages and compensate for the disadvantages of pure robust and stochastic approaches to make better decisions in complex domains under uncertainty.

1.2 Stochastic Optimization and Sequential Decision Problems

Data subject to uncertainty is usually represented via *random variables* – see [BL97, KWK94]. A random variable ξ does not take a value, but is instead *sampled* to obtain realizations, from a continuous or discrete set called *support* (i.e. the variable domain). A *probability distribution* defines how likely each value in the support is to be sampled. As already mentioned, stochastic optimization problems can be viewed as composed of multiple stages. At each stage, some uncertain elements are observed (i.e. one or more random variables are sampled), and some decisions must be made (i.e. some decision variables need to be assigned). The uncertainty is said to be exogenous if the distribution of the random variables in a stage does not depend on the decisions made in previous stages (e.g. weather conditions), and endogenous in the opposite case (e.g. recovering from an illness, while receiving cures).

The goal is to optimize a probabilistic performance measure (e.g. the expected problem cost), subject to both deterministic and probabilistic constraints. In the case of probabilistic constraints, the focus is on the reliability of the system, i.e., the system ability to meet feasibility in an uncertain environment. This reliability is expressed as a minimum requirement on the probability of satisfying constraints. Probabilistic constraints can be further divided on constraints over expectation (e.g. the expected stock of certain goods in a warehouse should be above a given level) or *chance* constraints (e.g. the probability that the stock is above a given level should be higher than a threshold). For more details, see e.g. [ZL11, LAGW08, Sah04].
Stochastic optimization needs a unified mathematical framework. In this perspective, [Pow16] describes and identify five common elements of potentially any stochastic optimization problem:

- State variable - It has the information needed to model a system at a given time. The elements of a state variable can include physical information or capture the probability distributions that describe the uncertainty.
- Decisions/actions/controls - These can come in a variety of forms (e.g. binary, discrete or continuous).
- Exogenous information - It describes new information that arrives over time from an exogenous (uncontrollable) source, which are uncertain to the system before the information arrives.
- Transition function - These are functions which describe how the state of the system evolves over time due to endogenous decisions and exogenous information. The transition function describes the evolution of all the state variables, and may exhibit a variety of mathematical structures (e.g. linear vs. nonlinear).
- Objective function - It is always assumed the presence of typically one metric (some applications have more) that can be used to evaluate the quality of decisions.

Generally, stochastic programs are more difficult than their deterministic counterparts, even if significant progress has been made towards their exact and approximate solution. Exact solution of deterministic equivalents of stochastic linear programs relies on decomposition: [BL97] reports the exact solution, on parallel computers, of stochastic linear programs with up to one million variables in their deterministic equivalents. Much larger problems are typically solvable by sampling-based rather than decomposition methods. These problems are solved using Sample Average Approximations.

1.3 Sampling and the Sample Average Approximation

With a few exceptions, the probability distributions of the random variables are approximated by drawing a finite number of samples [Sha13a]: this yields a collection of realizations referred to as *scenarios*. This sampling step can be done prior to the solution process in case of exogenous uncertainty, but must be performed at search time for endogenous uncertainty. Sampling and scenarios allow to tackle stochastic optimization via the Sample Average Approximation, [SP07, Sha13a]. In this approach, a set (i.e. a copy) of deterministic decisions is associated to each scenario, which allows to deal with expected values and chance constraints via summations and averages. To ensure meaningful solutions, it is important to add so-called *non-anticipativity* constraints to ensure that no decision is made with perfect knowledge of the future. In practice, if two scenarios share the same realization for the random variables in stages 1 to k, then their decisions for the stages 1 to $k+1$ must be identical. The SAA is a powerful and general method, but also very expensive from a computational point of view [Wal02]. For this reason, its application has been historically limited to offline, finite-stage, problems. This is in fact the context for which the SAA was originally developed, and the focus of many acceleration techniques such as the classical L-shaped method [LL93a].

An approach for stochastic programs with large sample spaces is the

use of Monte-Carlo sampling to generate i.i.d. realizations and approximate with a sample average approximating problem. Repeated solutions of the problem for different sample sizes along with statistical tests can provide approximate solutions together with estimates of the true optimal value. Monte-Carlo methods generally follow these steps:

1. Start from the statistical properties of possible inputs
2. Generate different sets of possible inputs with the above properties
3. Perform a deterministic calculation with these sets
4. Analyze statistically the results.

These methods are a subset of computational algorithms that use the process of repeated random sampling to make numerical estimations of unknown parameters. They allow for the modeling of complex situations where many random variables are involved. There are a broad spectrum of Monte Carlo methods (see [dMB14]), but they all rely on random number generation to solve deterministic problems.
Decision making under uncertainty traditionally has focused on a priori optimization which, as mentioned earlier, is orthogonal and complementary to online optimization. This is the case with stochastic programming, which is more concerned with strategic planning than operational decisions at the core of online algorithms.

1.4 Two-Stage Stochastic Programming

Applying the SAA to a two stage problem requires to determine a single set of decisions for stage 1, and one set of decisions (the so-called *recourse actions*) per scenario for stage 2 [Sha08]. In particular, the first-stage variables are those that have to be decided before the actual realization of the uncertain parameters. Subsequently, once the random events have presented themselves, further design or operational policy improvements can be made by selecting, at a certain

cost, the values of the second-stage, or recourse, variables. Traditionally, the second-stage variables are interpreted as corrective measures or recourse against any infeasibilities arising due to a particular realization of uncertainty. However, the second-stage problem may also be an operational-level decision problem following a first-stage plan and the uncertainty realization. Due to uncertainty, the second-stage cost is a random variable. The objective is to choose the first-stage variables in a way that the sum of the first-stage costs and the expected value of the random second-stage costs is minimized. The concept of recourse can be applied to linear, integer, and non-linear programming. This structure is depicted in Figure 2.1.

Figure 2.1: Two-Stage Stochastic Programming [HB09]

1.5 Multistage Stochastic Programming

The above formulations can be extended to multistage stochastic programs (see Fig. 2.2). This is often computationally intractable, and it states a problem that is the extension of decision trees to problems where a decision at certain time t is a vector that has to satisfy a set of constraints. [Sha08] has shown that the SAA method cannot be extended efficiently to multistage stochastic optimization problems: the number of required samples must grow exponentially with the number of iterations, which is typically large or infinite. Combinatorial solutions suffer from similar exponential explosion.

The recourse-based approach to stochastic programming requires to assign a cost to recourse activities that ensures feasibility of the second-stage problem. This approach allows infeasibilities in the second stage

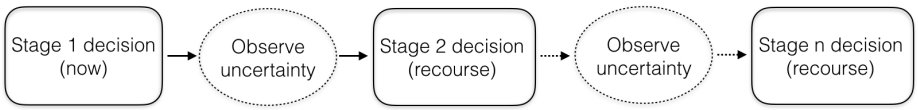

Figure 2.2: Multitage Stochastic Programming [HB09]

at a certain penalty. The objective is the minimization of expected recourse costs.

In the next sections we briefly introduce the concept of Stochastic Dynamic Programming that proposed how to rewrite time separable multistage stochastic optimization problems in the dynamic form, and Markov Decision Processes for sequential decision making. Then we focus on Online Stochastic Optimization.

1.6 Stochastic Dynamic Programming

The idea of moving from static optimization to a dynamic sequential one, allows to analyze the dynamic programming method [Bel58] which expressed the optimal policy in terms of an optimization problem with iteratively evolving value function (the optimal cost-to-go function).

The uncertainty is considered as part of the dynamic environment, generally considered a discrete-time system that evolves over N time periods (known as time horizon). It is assumed that, in a certain period k, the present state of the system is fully determined by its previous history and the objective function minimizes an additive cost function over the entire time horizon. It has been considered also a tail subproblem of minimizing the cost-to-go from time i to time N and the idea is that, regardless of how we arrived at state i, the remaining decisions must be optimal for the tail subproblem [Ros14].

Dynamic programming first solves all tail subproblems, then the original problem is solved at the last step of the process by utilizing the solutions of all tail subproblems.

It is necessary to use suitable algorithms to solve the tail problems (e.g. non-linear or other stochastic programming algorithms): as all tail subproblems must be solved by the algorithm, the procedure could be very computationally intensive. In practice, it is often necessary to limit the exponential growth of computational time and storage requirements in terms of the number of state and control variables. These difficulties have led to the development of several approximation techniques, including the approximation of the optimal cost-to-go function by that of a related simpler problem.

1.7 Markov Decision Processes

Markov Decision Processes (MDP) [Put14] is another fundamental model for sequential decision making. Generally, MDP consider a finite number of states and actions. At each time a state is observed and an action is executed, which incurs intermediate costs to be minimized (or rewards to be maximized). The cost and the successor state *depend only on the current state and the chosen action*. Successor generation may be probabilistic, based on the uncertainty we have on the environment in which the search takes place. For example, an action might sometimes fail to result in the desired target state, instead staying in the current state with a small probability.
MDP are often used to model sequential decision making, alternating between decisions and observations in which the uncertainty depends on the actions (endogenous). The biggest difference with stochastic optimization (i.e. the uncertainty is exogenous) pushed [HB09] towards the definition of a variant of MDP where the uncertainty is exogenous (i.e. Markov Chance-Decision Processes (MCDP)) whose benefits for stochastic optimization are computational: they can be tackled online using anticipatory algorithms [HB09]. Indeed, because the uncertainty is exogenous, MCDP naturally allow for the anticipatory relaxation that removes the interleaving of decisions and observations and is expressed in terms of deterministic optimization problems. The anticipatory algo-

rithms can thus exploit the anticipatory relaxation on scenarios of the future in order to make better decisions online.

2 Towards Online Stochastic Optimization

When we are dealing with a problem with more than two stages, optimizing at run time provides the opportunity to adapt the solutions to unexpected events (since those can be observed) and to reduce the computational cost (since there is no need to plan for every possible outcome). This line of reasoning is at the basis of stochastic online optimization that, due to the frequent presence of tight time limit on the solution process, has been traditionally tackled via heuristics.
Ideally, an offline optimization would compute an optimal policy to an accurate model of the application. However, such models need to account for a huge amount of rare events that induce high computational costs. If the offline optimization is the only option, we need to simplify the model by obtaining optimal or near-optimal solutions to an approximated problem.
Since online algorithms can react to external events or anticipate the future, often uncertainties are better handled online. We can observe that the positive aspect to approach online optimization is that it avoids the need to search for policies in huge search space. On the other hand, the price to pay is the need to optimize online with the possibility of not satisfying tight time constraints. Moreover, the synergy of offline optimization (to compute robust architectures) and online optimization (to use these architectures adaptively) has the potential to find high-quality solutions to the real problems and it is the aim of this thesis.

2.1 Online Stochastic Optimization

In general, online algorithms use only the revealed inputs and past decisions to take the next decision, and a competitive analysis [FW98] is

used to analyze their performance. It means that the online algorithm is compared to an offline algorithm for the same problem. Online *stochastic* combinatorial optimization algorithms use a black-box to sample scenarios of the future and they exploit past *and future* information to take their decisions. The goal is to maximize the expected profit (or minimize the expected cost) of the online algorithm. In recent years, the availability of improved algorithms has enabled the application of sampling-based algorithm also in an online setting: these are often referred to as *anticipatory algorithms*, many of which received excellent coverage in [HB09].

2.2 Online Anticipatory Algorithms

Generally speaking, online anticipatory algorithms need to be run at each stage and rely on scenarios to obtain approximate information about the future: this enables significant improvements in terms of quality, but comes with a substantial computational cost that must be carefully managed.

For example, the EXPECTATION algorithm [BVH04a] attempts to reduce the solution time by optimizing each scenario independently (and therefore as a deterministic problem), for all possible decisions; the method then selects the decision which maximizes the expected profit. The CONSENSUS algorithm [BVH04d] improves over this scheme by solving a deterministic problem per scenario. Every time a decision for the current stage is picked as optimal in one of those problems it receives a votes; once the process over, the algorithm chooses the decision with the most "votes".

The technique employed by CONSENSUS has some adverse effects on the solution quality, which are addressed in REGRET algorithm [BVH04b] by extracting more information from each solved problem; this leads to a more reliable selection of the optimal decision for the current stage. The AMSAA method [MVH08] instead hybridizes SAA and Markov Decision Processes techniques to improve the solution quality at the

expense of the computational cost. All the anticipatory online algorithms mentioned so far are applicable only to problems with discrete, enumerable, decisions.

There is always a trade-off between the computation cost and the quality and robustness of the provided solution. This trade-off is the primary object of investigation in [MH07], and can be tuned by adjusting the number scenarios and the so-called look ahead horizon, i.e. the number of future stages that are taken into account in each scenario. More in general, the method for generating the scenarios can be adapted to the given problem and the user goals, as described for example in [KW03a].

As mentioned before, the literature on optimization under uncertainty has focused on offline problems that usually rely on sampling (yielding a number of *scenarios*) to obtain a statistical model of future uncertainty. Robust solutions can be obtained by building one copy of the decision variables per scenario, and linking them via *non-anticipativity constraints* (decisions based on the same observations should be identical): SAA [KSHdM02] provides convergence guarantees under reasonable assumptions, and can substantially outperform myopic optimization.

More recently, improvements in the solution techniques and computational power have enabled the application of *online anticipatory algorithms*, which proved very effective at finding robust, high quality, solutions as uncertainty slowly reveals itself.

Online anticipatory algorithm typically rely on scenario sampling to estimate the possible developments for a fixed number of future steps, known as *look-ahead horizon*. Larger sample sizes result in higher accuracy, but also in more and bigger (possibly NP-hard) problems to be solved. This is a strong limitation, since in many practical cases online decision must be produced within strict time limits. Considerable research effort has therefore focused on improving the efficiency of these algorithms. For example, the CONSENSUS and REGRET al-

gorithms from [BVH04c] both attempt to reduce the number of problems w.r.t. the earlier EXPECTATION approach. Computational studies such as [BEY05, MVH07] aim at characterizing the algorithm sensitivities to their design parameters (such as the number of sampled scenarios and the look-ahead horizon). The approaches from [JL96, PDM12, LTY13, DFLM19] attempt instead to reduce the number of scenarios by increasing their relevance, and in particular by taking into account past observations while sampling. We also focused on this aspect, in particular in the second part of this thesis, to define general methods in order to manage the cost/quality trade-off of online stochastic anticipatory algorithms, taking advantage of exploiting the existence of an offline phase with some useful offline information (e.g. forecasts or historical data).

3 Integrated Offline/Online Decision-Making in Complex Systems

The need to account for multiple future developments makes stochastic optimization incredibly challenging, which explains how approximate (sampling-based) methods and heuristics are the most popular solution techniques. Due to such a complexity, the applicable approaches depend on the temporal granularity of the decisions to be made. Long-term "strategic" decisions (which are often very impactful) are typically solved via expensive, but more accurate, sampling-based approaches. Short-term "operational" decisions often need to be made over multiple steps, within a short time frame: they are commonly addressed via polynomial-time heuristics, while more advanced sampling-based methods are applicable only if their computational cost is carefully managed.

A classical example, to better understand the motivations and the context, is a real-world management system which involves the planning,

scheduling and control of activities with different temporal granularity of the decisions to be made. Traditionally, theoretical approaches in literature have mainly focused on the scheduling phase assuming a static and deterministic environment. However, in practice, there is a need to consider uncertainty in order to prevent incurring costs due to unexpected events with a negative impact on project milestone completion times. In the perspective of time horizons and objectives of decisions, project and system management decision making can be subdivided into three levels [HHLW07, DDH+07]. The strategic level is concerned with long-term decisions made by top level management (e.g. major capital investments and project financing). On the tactical level, decisions are made regarding project acceptance. Finally, the scheduling decisions are made at the operational level. This involves the allocation of specific resource units to project activities and the scheduling of those activities in time together with reacting to schedule changes when needed. It is important to focus on the interdependencies between these obviously related decision levels by also taking into account, at every decision level, the source of uncertainty.
As shown in different real-world and literature examples, the distinction between offline and online problems is somewhat blurry: in this thesis, we will refer as "online" to *problems that need to be solved repeatedly over time*, with the outcome of each solution attempt affecting the subsequent ones. In practice, online problems often need to be solved within strict time limits, while this requirement is relaxed for offline problems.

3.1 Motivating Examples

In this section, we review some real world use cases that are typically solved via either offline or online models, while in fact they are *integrated offline/online problems*.

- *Energy Management Systems (EMS)* are key components of the electrical grid that maintain its stability both by shifting con-

sumption (over time) and routing power flows from the available generators. EMS need to tackle a very challenging problem, due to the progressive shift towards decentralized generation, the strong penetration of (uncontrollable and stochastic) Renewable Energy Sources (RES), and the integration of flexible (deterministic) energy systems. In practice, the load shifts must be planned offline (the day ahead) and the power flow balance should be maintained online (e.g. hour by hour), so as to minimize the costs (see [MCM+13, CBRJ15]).

- In *transportation systems*, a central role is played by the Vehicle Routing Problem and its variants [TV02], which consists in establishing the paths for a set of vehicles to serve a set of customers. In a real world setting, many aspects (e.g. customer demands and travel times) are also subject to uncertainty [MNP14]. Several transportation companies focus on assigning customers to smaller scale operators (offline), which are then in charge of choosing the routes (online).

- In *project scheduling* the goal is to generate a feasible schedule that optimizes some performance metric (i.e. the project duration), in presence of limited resources. This schedule can serve as a basis for planning external activities such as material procurement, preventive maintenance and delivery of orders to external or internal customers. During execution, project activities are subject to considerable uncertainty that may lead to schedule disruptions [HL05]. A disrupted schedule incurs higher costs due to missed deadlines, resource idleness, higher work-in-process inventory and possible frequent rescheduling. Like in the previous examples, it is possible to plan project activities offline and then to use online algorithms to improve (online) solutions as the elements of uncertainty reveal themselves.

- In *reservation systems* requests arrive online and must be dynamically allocated to limited resources in order to maximize profit [VHBV06]. Example include hotel of flight booking systems, which are both subject to considerable uncertainty in the real world. Once again, a base reservation plan is usually devised offline, but it then needs to be integrated with an online dynamic system to cope with unexpected disruptions.

All such problems feature both offline and online phases, which are typically solved in isolation, despite being strongly interconnected. In this thesis, we will show that a tighter integration between the two phases can lead to substantial improvements: this will be done via an empirical evaluation using the first two examples (energy management and transportation system) as case studies.

3.2 Offline/Online Models

In optimization under uncertainty *a suitable representation for the uncertainty must be found* and (except in rare cases) some technique must be used to *trade estimation accuracy for a reduction of the computation time*. As already said, data subject to uncertainty can be often represented via *random variables in a multi-stage decision system*. After taking the decisions for a stage a random event occurs, i.e. some of the random variables are instantiated, and the decisions for the next stage must be taken, and so on.
As already mentioned, it is common to use *sampling* to approximate the probability distribution of the random variables [Sha13b]. Sampling yields a number of *scenarios*: then, a single set of decisions is associated to the current stage, while separate sets of decisions are associated to each scenario in the next stage. More scenarios result in a better approximation, but a larger computation time. Looking more than one

stage ahead also improves the estimation quality, but it requires to repeat the procedure recursively, with major impacts on the solution time.

There is a delicate trade-off between speculating vs. waiting for the uncertainty to be resolved [KW03b]. This leads to an informal (but practical) distinction between *offline and online problems*: online algorithms require to make decisions over time as the input is slowly revealed and delaying decisions can either increase the costs or be impossible due to constrained resources.

To summarize from the previous sections, offline problems are often solved via exact solution methods on approximate models with limited look-ahead, e.g. via two-stage scenario-based approaches where both the first-stage and second stage variables are instantiated, or via decomposition based methods [LL93b].

Online problems are often tackled in practice via greedy heuristics, but more rigorous and effective anticipatory algorithms are also available as long as the temporal constraints are not too tight, e.g. the AMSAA algorithm from [HB09, MVH08]. Similarly to offline approaches, online anticipatory algorithms take decisions by solving deterministic optimization problems that represent possible realizations of the future. They address the time-critical nature of decisions by making efforts to yield solutions of reasonable quality early on in the search process.

In this thesis, we are interested in optimization problems with both an offline and an online component. Formally, we focus on n-stage problems where the first-stage decisions are "strategic" (and can be taken with relative leisure), while the remaining $n-1$ stages involve "operational" decisions (with tighter temporal constraints). For the methods defined in the first part of the thesis, we made the assumption that a greedy heuristic, based on a convex optimization model, is available for the online part. *As a baseline, we deal with the offline decisions by collapsing the $n-1$ on-line stages into a single stage*, and then obtaining via sampling a classical two-stage model. The online part is tackled with the original heuristic. This results into a relatively efficient

approach, but yields solutions of limited quality.

4 Optimization Models under Uncertainty for Energy Management Systems

There is a wide range of problems in energy systems that require making decisions in the presence of different forms of uncertainty which pervades optimization problems in the energy sector (e.g. unit commitment, renewable energy production, market prices,...).
Energy systems, consisting of a strong penetration of renewable energy resources, are subject to uncertainty. This situation is extremely common but also very complex to manage and model, since the integration of renewable sources must be adequately treated in order to manage uncertainty and avoid compromising the operational reliability of the energy system.

4.1 Distributed Generation and Virtual Power Plants

The progressive shift towards decentralized generation in power distribution networks has made the problem of optimal Distributed Energy Resources (DERs) operation increasingly constrained. This is due to the integration of flexible (deterministic) energy systems with the strong penetration of (uncontrollable and stochastic) Renewable Energy Sources (RES). The integration of these resources into power system operation requires a major change in the current network control structure. This challenge can be met by using new and different concepts like Virtual Power Plant (VPP), which is based on the idea of aggregating the capacity of many DERs, (i.e. generation, storage, or demand) to create a single operating profile and manage the uncertainty.
A VPP is one of the main components of future smart electrical grids, connecting and integrating several types of energy sources, loads and storage devices. A typical VPP is a large industrial plant with high (par-

tially shiftable) electric and thermal loads, renewable energy generators and electric and thermal storages (see Figure 4.1).
In a virtual power plant Energy Management System (EMS), the load shifts can be planned offline, while the energy balance should be maintained online by managing energy flows between the grid, the loads, the renewable and traditional generators, the storage systems.
This makes a VPP management a good candidate for grounding our approaches. Based on actual energy prices and on the availability of DERs, the EMS of a VPP decides:

1. how much energy should be produced;
2. which generators should be used for the required energy;
3. whether the surplus energy should be stored or sold to the energy market;
4. the load shifts planned offline.

Optimizing the use of energy can lead to significant economic benefits, and improve the efficiency and stability of the electric system (see e.g. [PBBA+11a]).

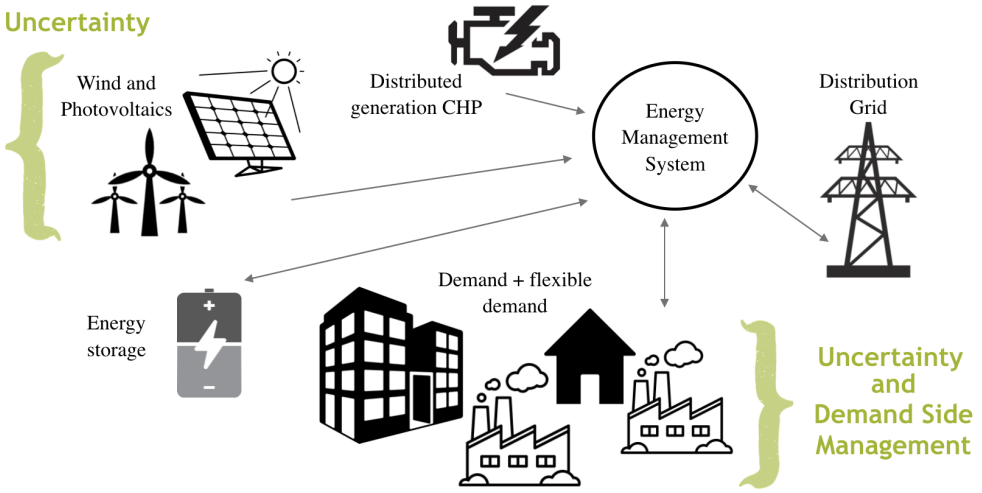

Figure 2.3: A typical Virtual Power Plant

4.2 Optimization Techniques

Electric systems are increasingly converging towards a full integration of renewable sources into the electricity grid. The spread of Distributed Generation (DG) involves the emergence of a series of critical issues in the management of electrical systems and the transmission network, together with the need for new management criteria and innovative technical solutions to contain the costs of running the entire system. The evolution of networks is thus pushed towards greater flexibility, efficiency and reliability. In this context, the concepts of Smart Grid and Virtual Power Plant are born to effectively and efficiently integrate distributed generation.

During the last decade several new concepts of energy planning and management such as decentralized planning, energy conservation through improved technologies, integrated energy planning, introduction of renewable energy sources and energy forecasting have emerged. The different types of models such as energy planning models, energy supply demand models, forecasting models, renewable energy models, emission reduction models, optimization models have been reviewed and presented in [RBG11, BZ11, JI06].

The problem of scheduling and planning of Distributed Energy Resources is typically addressed by introducing a local Energy Management System [PBBA+11a], which coordinates power flows from generators, controllable loads and storage. The goal is to minimize electricity generation costs and avoid the loss of energy produced from renewable energy sources in aggregates like Virtual Power Plants.

The potential applications of VPP has been recognized in recent literature. For example, [AP12] shows that the advance of DER in the commercial and regulatory structure of electricity markets in course of liberalization has created opportunities for decentralization of the role of traditional power utilities. VPPs are one of the main components of intelligent electrical grids of the future, connecting and integrating several types of power sources (both renewable and non-renewable),

storage and energy loads to operate as a unique power plant. The heart of a VPP is an EMS which coordinates the power flows coming from the generators, controllable loads and storages. In [LPR09] an EMS for controlling a VPP is presented, with the objective to manage the power flows for minimizing the electricity generation costs, and avoiding the loss of energy produced by renewable energy sources.

DER aggregation can effectively couple traditional peak electrical plants by supporting them with the flexible contribution of consumers to the overall efficiency of the electric system. From this perspective, the EMS of a VPP can develop Demand Side Management (DSM) mechanisms to modify temporal consumption patterns. DSM can provide a number of advantages to the energy system and focuses on utilizing power saving mechanisms, electricity tariffs, and government policies to decrease the demand peak and operational costs instead of enlarging the generation capacity. As an example, [PBBA+11a] proposed an Energy Management System for a renewable-based microgrid with online signals for consumers to promote behavior changes.

Optimization techniques such as Demand Response (DR) can bridge the gap between production and real consumption in the energy management of complex energy systems (i.e. Virtual Power Plant) to reduce operating costs. These techniques can increase energy efficiency by moving part of the energy consumption during non-peak hours [PD11]. In addition to environmental benefits, DR mechanisms provide end users with the opportunity to reduce electricity costs by responding to market prices. To this end, optimization models such as [DFLM17] have been developed to support political decision makers (local governments) and economics (managers) in defining sustainable business models and energy tariffs. The system has been integrated into an ICT services platform to promote energy efficiency.

The management of next-generation energy systems requires accurate models and data-driven approaches. These models can be categorized into three groups: descriptive, predictive and prescriptive. The descrip-

tive models aim to provide an accurate and interpretable view of the state of the system, or to determine the causes of certain events (diagnostics). Predictive models include techniques for estimating the possible evolution paths of the system and their probability. Finally, the prescriptive models attempt to quantify the effect of possible decisions, to support the manager or users of the system in choosing the best course of action. Artificial Intelligence techniques can be applied to all three levels, to improve the efficiency of the infrastructure, its reliability, and resilience with respect to unexpected events.

During the last years, optimization techniques in the energy sector are focused on the integration between predictive and prescriptive level, in particular on the use of predictive models extracted from data (e.g. through Machine Learning) within optimization processes and decision support. The approach called Empirical Model Learning (EML) [LMB17] allows the application of declarative optimization methods to complex systems. To design real-world decision support systems it is necessary a good combinatorial optimization model that takes into account the uncertainty. Often enough, accurate predictive models (e.g. simulators) can be devised, but they are too complex or too slow to be employed in combinatorial optimization. EML is based on the idea of using a Machine Learning model to approximate the input/output behavior of a system that is hard to model by conventional means; embedding such Empirical Model into a Combinatorial Optimization model. The emphasis of EML is mostly on the techniques to perform the embedding. These should be designed so that the optimization engine can exploit the structure of the empirical model to boost the search process. The range of potential applications of EML is quite vast and includes: (1) applying Combinatorial Optimization to Complex Systems (in the proper sense), or systems that are too complicated to obtain an expert-design; (2) enabling prescriptive analytics by taking advantage of a pre-extracted predictive analytics model; (3) enable indirect interaction between a high-level optimizer and a lower-level optimizer

(whose approximate behavior can be captured via Machine Learning). In this perspective, it is possible to use EML to enable multi-level optimization and therefore optimization over large scale systems such as large distributed energy management systems.

The use of optimization methods, such as Mathematical Programming, at the heart of an Energy System can allow a rationalization of energy use, minimizing costs, losses and improving environmental impacts. However, every mathematical model for an Energy System must inevitably come to terms with inevitable sources of uncertainty, linked to intermittency and partial unpredictability of renewable energy sources.

Making decisions under uncertainty pervades the planning and operation of our energy system [WF03]. Even if optimization techniques have a long tradition in supporting planning and operational decisions in the energy sector, the recent literature highlights the need for increasing both the scope and the granularity of the decisions, including new factors like distributed generation by renewable sources and uncertainty.

Both the most popular methods to deal with uncertainty in mathematical programming (i.e. robust optimization and stochastic programming) have been widely applied in energy systems [RSJ17, ZZL+13a, JPK15a]. One of the most used assumption is that the distribution of future uncertainty is available for sampling, e.g. thanks to historical data and/or predictive models. In particular, the assumption that the distribution of future uncertainty is independent of current decisions is present in a variety of applications [HB09].

The integration of renewable sources must be adequately addressed so as to manage uncertainty and to avoid affecting the operational reliability of a power system. Unit commitment (UC) is a critical decision process, which can be formalized as the problem of deciding the outputs of all the generators to minimize the system cost. The main principle in operating an electrical system is to cover the demand for electricity at all times and under different conditions depending on the season,

weather and time, and by minimizing the operating cost. The deterministic formulation of this problem may not adequately account for the impact of uncertainty.
For this reason, different approaches are used to manage UC under uncertainty [JPK15b]:

1. *Stochastic UC*, which is based on probabilistic *scenarios*. The basic idea is to find optimal decisions taking into account a large number of scenarios, each representing a possible realization of the uncertain factors. Stochastic UC is generally formulated as a two-stage problem[ZZL$^+$13b] that determines the generation schedule to minimize the expected cost over all of the scenarios, while respecting their probabilities. The approach usually requires high computational cost for simulations.

2. *Robust UC* formulations, which optimize assuming a well-defined range for the uncertain quantities, instead of taking into account their probability distribution. The range of uncertainty is defined by the upper and lower bounds on the net load at each time period. Instead of minimizing the total expected cost as in stochastic UC, robust UC reduces the worst-case costs for all possible results of uncertain parameters [ZWL15].

3. *Hybrid models* have been proposed in recent years to combine the advantages and compensate the disadvantages of pure robust and stochastic approaches [ZG13].

The assessment of uncertainty in the modeling of distributed energy systems has received considerable attention in recent works that apply machine learning techniques for forecasting flexibility of VPP. Many studies have been done on the residential sector using support vector regression and neural networks [ENP12, JSCT14] and some methods present promising results however it seems unlikely they may be implemented in real life in particular in the industrial sector.

In many practical cases, the problem is further complicated by the need to take both complex strategic decisions (e.g. consumption planning for a period of time) and quick operational decisions (e.g. routing of energy flows). Long-term strategic objectives must coexist with medium and short-term operational objectives in the energy model decision-making process.
We developed a series of methods for integrated offline/online optimization in the presence of uncertainty [DFLM18c, DFLM18b] and we tested them over a VPP energy management system. These methods are of general applicability, and have been shown to provide considerable benefits in terms of quality and robustness of the solutions, in different simulated contexts.
These methods, integrated with Machine Learning, are going to be also used to develop innovative architectures for energy systems that allow to consider:

1. an optimized local energy management in industrial and tertiary contexts, with a high degree of resilience with respect to elements of uncertainty, such as (e.g.) renewable energy sources, deviations from the estimated consumption plans
2. a decision-making process for the core of an energy management system based on different temporal granularities, with the possibility of integration of offline and online decisions
3. ability to manage multiple objectives, because the behavior of the energy system can be evaluated according to different metrics (for example of an economic nature, related to the reliability and stability of the system, or relative to environmental aspects).

Chapter 3

Offline/Online Integration in Optimization under Uncertainty

1 Introduction

Optimization problems under uncertainty can be seen as a sequence of multiple stages, such that at each stage part of the uncertainty is revealed and some decisions must be made. Such decisions are irrevocable and made without full knowledge of the future: they should therefore account for multiple (ideally all) possible outcomes, and optimize a *probabilistic* performance measure (e.g.the expected value of a relevant cost metric).

1.1 Strategic and Operational Decisions

The need to account for multiple future developments makes stochastic optimization incredibly challenging, which explains how approximate (sampling-based) methods and heuristics are the most popular solution techniques. Due to such a complexity, the applicable approaches depend on the temporal granularity of the decisions to be made. Long-

term "strategic" decisions (which are often very impactful) are typically solved via expensive, but more accurate, sampling-based approaches. Short-term "operational" decisions often need to be made over multiple steps, within a short time frame: they are commonly addressed via polynomial-time heuristics, while more advanced sampling-based methods are applicable only if their computational cost is carefully managed. We will broadly refer to the first class of problems (and solution approaches) as *offline* and to the second as *online*.

1.2 Model Description and Motivations

In this chapter, we move from the observation that many practical application scenarios require to make *interdependent offline and online decisions*. For example, we may need to define a daily production schedule for an industrial plant, and then manage its power supply on a hour by hour basis; or we may assign customers to vehicles for delivering goods, and then adjust their routes dynamically as the traffic conditions reveal themselves over time. The simplest approach to tackle such problems is to deal with the offline and online phase separately, respectively (e.g.) via a sampling-based method and a heuristic. However, we will show that *substantial improvements can be obtained by treating the two phases in an integrated fashion*.

As a baseline, we consider an approach that deals with offline decisions via a sampling-based method, and with online decisions via a greedy heuristic. This baseline is not problem specific, instead we simply assume that: 1) the uncertainty is exogenous; 2) a two-stage stochastic optimization model is used for the offline phase; 3) the online heuristic can be stated as convex optimization problem. We then show how to improve the baseline in different directions, each altering either the offline or the online component of the solution process, so that the two play better together. All our methods are applicable under the same (general) assumptions as the baseline. *We believe our techniques represent a significant step toward integrated offline/online optimization*.

To test our methods, we ground them on two case studies, matching the examples mentioned earlier: 1) an energy system management problem, where load shifts are planned offline (the day ahead) and power flows must be controlled online (e.g. hour by hour); and 2) a Vehicle Routing Problem where customer are assigned offline, but the routes can be chosen online (i.e. based on the uncertain travel times). The first problem features a continuous (and hence non-enumerable) decision space, while the second has pure discrete decisions. In our experiments, all the proposed methods significantly improve over the baseline in terms of solution quality. While the computation cost is always higher than the baseline, each approach hits a different trade-off in terms of offline and online solution time.

This chapter is structured as follows: Section 2 describes the starting baseline model which is designed to be representative of this state of the art. Section 3 describes in details our proposed methods (as improvements of the baseline model) by pointing out the importance of both the offline and the online part for each method.

2 Baseline Model: Formal Description

We can now proceed to describe our baseline method. Historically, methods such as stochastic optimization – see [SP07, BL97, KWK94] – have been used for the offline phase, while the online phase has often been tackled via simple, non-anticipatory, heuristics. Our baseline method is designed to be representative of this state of the art. In particular, we use a sampling-based model for the offline decision that is already capable of taking into account the existence of the online phase, albeit in a limited fashion. For the online phase itself, we use instead a fast greedy heuristic.

We assume *exogenous uncertainty*, and that the overall management system is composed by two macro steps: *the offline decisions are made by a two-stage stochastic optimization model*, based on sampling and

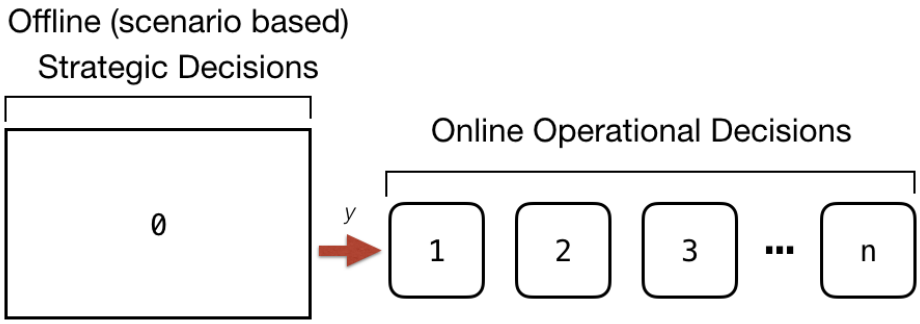

Figure 3.1: Baseline Offline and Online Integration [DFLM18a]

scenarios. The second step is an online algorithm, implemented within a simulator, that tries to make optimal online choices, by building over the offline decisions. We make the assumption that *the online algorithm is based on a convex optimization model*. We allow such model to have some configuration parameters: for example, the parameters may either specify the cost of each action, or may represent constants used for score computation.

We view mixed offline/online problems as on n-stage problems where the first-stage decisions are *strategic* (and can be taken with relative leisure), while the remaining n - 1 stages involve *operational* decisions (with tighter temporal constraints).

In the model descriptions, y will represents the offline decisions; x^k will represent the online decisions for stage k; s^k (resp. ξ^k) will represent the system state (resp. the uncertainty) revealed at the beginning (resp. the end) of stage k. All variables are assumed to be vector-valued; they can be either continuous or discrete, and have either finite or infinite domain.

We will refers as $\mathcal{F}(y, x^k, s^k)$ to the cost incurred at stage k for taking decisions x^k. The cost directly associated to the offline decisions is instead referred to as $\mathcal{F}_o(y)$. Therefore, the total cost for a single run over all the stages is given by:

$$\mathcal{F}_o(y) + \sum_{k=1}^{n} \mathcal{F}(y, x^k, s^k) \tag{3.1}$$

The transition from the state in stage k to the state in stage $k+1$ is defined by means of a *transition function* T, i.e.:

$$s^{k+1} = T(y, x^k, s^k, \xi^k)$$

where it can be seen that the effect of the uncertainty (i.e. the random variable) is encoded in the state.
This Baseline will be improved in Section 3, via three broad ideas: 1) improving the online heuristic by adding an anticipatory component; 2) making the offline solver aware of the online heuristic and its limitations; 3) tuning the parameters of the online heuristic to alter its behavior. The first idea is closely related to existing online anticipatory algorithms (e.g. EXPECTATION); the second and third ideas exploit the mixed nature of the problem to enable improvements via a deeper integration of the offline and online phases. We formalize our methods to propose general approaches that can be applied to different real world use cases, as long as a few basic assumptions are satisfied.

2.1 Flattened Problem

Before introducing the model for the offline phase, it is useful to discuss a common approximation technique employed to reduce the computational cost of solving a multi-stage problem.
Let Ω be a set of scenarios ω for $\xi = (\xi^0, \dots \xi^{n-1})$. Given a single scenario ω, it is possible to collapse the constraint and cost of each stage to obtain a *flattened (online) problem*:

$$\min \sum_{k=1}^{n} \mathcal{F}(y, x_\omega^k, s_\omega^k) \qquad (\mathbf{PF})$$

$$\text{s.t. } e(y, x_\omega^k, s_\omega^k) = 0 \qquad \forall k = 1..n \qquad (3.2)$$

$$g(y, x_\omega^k, s_\omega^k) \leq 0 \qquad \forall k = 1..n \qquad (3.3)$$

$$s_\omega^{k+1} = T(y, x_\omega^k, s_\omega^k, \xi_\omega^k) \qquad \forall k = 1..n-1 \qquad (3.4)$$

where x_ω^k/s_ω^k/ξ_ω^k are the online decisions/state/realizations for stage k in

scenario ω. Functions e and g are vector-valued in general and define the constraints for each stage.
Since **PF** assumes the availability of all ξ_ω^k values, it is effectively a clairvoyant approach, due to the lack of non-anticipativity constraints. In the online optimization literature the flattened problem is better known as the offline problem [HB09]: we adopt a different name to avoid ambiguity with the actual offline phase.
Note that the flattened problem is obtained by collapsing *online* stages, for which we have made a convexity assumption. This implies that g must be convex and e linear. From a computational standpoint, this also means that **PF** is largely convex itself, and that its complexity depends heavily on the properties of the state transition function. If T is linear, then the flattened problem will be convex and relatively easy to solve. Non-linear transition functions are conversely much harder to handle.

2.2 Offline Problem

As a baseline to deal with the offline decisions we consider a two-stage stochastic optimization problem obtained by instantiating **PF** once per scenario:

$$\min \mathcal{F}_o(y) + \frac{1}{|\Omega|} \sum_{\omega \in \Omega} \sum_{k=1}^{n} \mathcal{F}(y, x_\omega^k, s_\omega^k) \qquad (\mathbf{PO})$$

$$\text{s.t. Eq. (3.2)} - \text{(3.4)} \qquad \forall \omega \in \Omega$$

$$s_\omega^1 = T_o(y, \xi_\omega^0) \qquad \forall \omega \in \Omega \qquad (3.5)$$

$$y \in \mathcal{Y} \qquad (3.6)$$

where we recall that $\mathcal{F}_o(y)$ represents the cost that depends directly on the offline decisions. The remainder of the cost function is given by the Sample Average Approximation of the expected cost of the subsequent stages. The function $T_o(y, \xi_\omega^0)$ determines the initial state for the online stages, based on the value of y and on the uncertainty revealed at the end of the offline stage (i.e. ξ_ω^0). Finally, $\mathcal{Y}$ is the feasible space for

the offline decision variables y. We make no special assumption on $\mathcal{Y}$, $\mathcal{F}_o(y)$, and $T_o(y, \xi_\omega^0)$, meaning that even when the flattened problem is convex the offline problem may be NP-complete (or harder). Still, the fact that the problem is solved offline makes its complexity less critical.

2.3 Online Heuristic

Since we assume that the online heuristic can be modeled as a parametric convex optimization problem, we have that:

$$\begin{aligned} &\min f(y, x^k, s^k; \alpha^k) && (\mathbf{PH}) \\ &\text{s.t. } e(y, x^k, s^k) = 0 && (3.7) \\ &\quad g(y, x^k, s^k) \leq 0 && (3.8) \end{aligned}$$

where f is the cost function with parameter vector α^k, while e and g are the same constraint functions appearing in **PF**.
Note that the objective function f is not in general the same as the actual cost $\mathcal{F}(y, x^k, s^k)$ incurred at stage k: using a modified cost function is actually a common technique employed by domain experts to control the behavior of a heuristic.
Problem **PH** is general enough to capture heuristics of practical interest, such as shortest link selection in routing, or Priority Rule Based scheduling (aka List Scheduling): in this cases, the constraints define the available actions and the cost function allows to rank them.

3 Improving Offline/Online Integration Methods

The biggest drawback of the approach from Section 2 is that using the flattened problem to estimate the effect of the offline decision on the future is equivalent to assuming the availability of perfect information. However, the greedy heuristic employed for the online phase is instead completely myopic. *This creates a discrepancy between the estimates*

made by the offline solver and the capabilities of the online solver, which intuitively should have an adverse effect on the performance on the overall problem.

Such a discrepancy can be addressed by following two strategies. First, we can *improve the online solver by adding some anticipatory capabilities*. Second, we can *make the offline solver explicitly aware of the limitations of the online approach*. Both methods have the effect of bridging the gap between the tools used in the offline and online phase. They are also not mutually exclusive, and in fact one of the approaches we present acts in both directions.

Probably the most natural way to improve online decision making consists in replacing the greedy heuristic with a sampling-based anticipatory algorithm: this is the key idea in our ANTICIPATE method (see Section 4). However, increasing the computational load of the online phase may not a good idea when stringent time constraints exist. In such a situation, it may be better to improve the greedy heuristic by simply adjusting its parameters. This is the main idea in the TUNING approach: this maintains the efficiency of the original greedy heuristic, at the price of a computationally expensive parameter tuning process, which is however performed offline (see Section 5).

Shifting our attention to the offline decision, we can mitigate the discrepancy by translating the online greedy heuristic as a set of constraints, which can be injected in the offline model **PO**. This techniques leads to our ACKNOWLEDGE method (see Section 6). Interestingly, we show in Section 7 that the approach can be combined with parameter tuning to achieve even deeper integration: this idea is explored in our ACTIVE method (see Figure 3.2).

4 ANTICIPATE

We can derive a sampling-based anticipatory algorithms for the online phase via the same method employed for the offline problem in our

Technique	What to improve?	Where is the effort?
Anticipatory Heuristic	Online	Online
Tuning Heuristic	Online	Offline
Heuristic Aware Offline Decisions	Online	Offline

Figure 3.2: General Building Block Techniques.

baseline, i.e. instantiating **PF** for the remaining stages (and for all samples). Formally, let h be the index of the current stage, then we consider:

$$\min \mathcal{F}(y, x^h, s^h) + \frac{1}{|\Omega|} \sum_{\omega \in \Omega} \sum_{k=h+1}^{n} \mathcal{F}(y, x^k_\omega, s^k_\omega) \qquad (\mathbf{PA})$$

$$\begin{aligned} \text{s.t. Eq. } (3.7, 3.8) \quad & \text{– } \textit{online problem constraints} \text{ –} && \text{for stage } h \\ \text{Eq. } (3.2, 3.3) \quad & \text{– } \textit{flattened problem constraints} \text{ –} && \text{for } k > h \\ \text{Eq. } (3.4) \quad & \text{– } \textit{state} \text{ –} && \text{for } k \geq h, \text{ with } s^h_\omega = s^h \text{ and } x^h_\omega = x^h \end{aligned}$$

The offline decisions are taken like in the baseline, i.e. by using **PO**. This first approach, referred to as ANTICIPATE, improves the accuracy of the online component at the expense of its solution time. In particular, the need to take into account state transitions may make **PA** NP-hard even if the constraints for each online stage are convex.

PA has the same semantic as the EXPECTATION algorithm, except that this is done by solving a single optimization problem rather than one problem for each scenario and for each possible decision in the current stage. As a main drawback, the problem that our method needs to solve may be considerably larger (as it takes into account all scenarios simul-

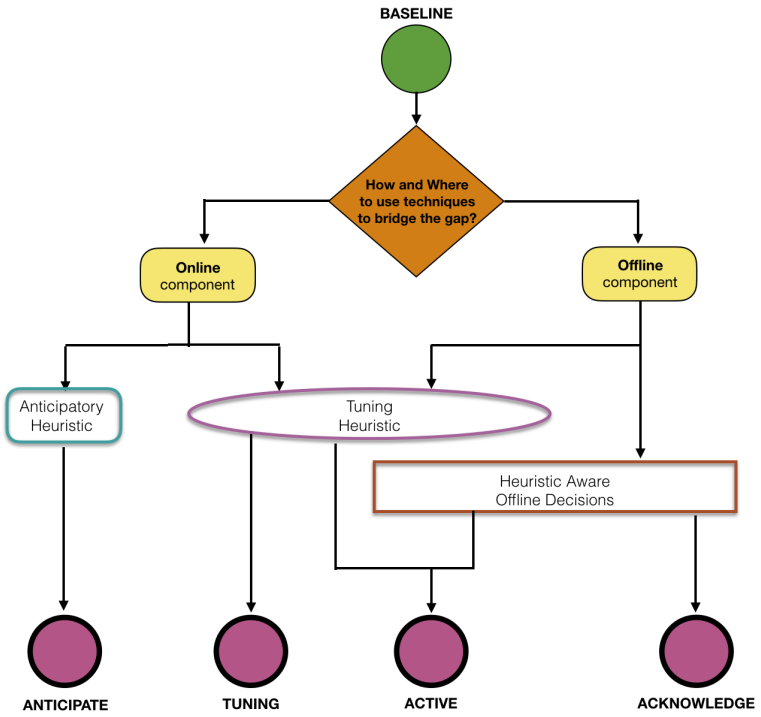

Figure 3.3: Techniques and component to generate our methods.

taneously): in many cases, this trait makes our approach less efficient than EXPECTATION (and therefore than CONSENSUS and REGRET).

However, there are two important practical cases where our approach has a substantial advantage. First, when the decision for each stage consists of multiple "components" (e.g. choosing subsets of items in a knapsack problem) the number of potential alternatives may grow very large. In such a situation,the EXPECTATION algorithm may become rather costly (due to the need to enumerate all subsets), while CONSENSUS and REGRET may have difficulties in obtaining a valid estimate of the expected impacts (since costs cannot be readily ascribed to individual items). Second (and more importantly), when the decision space is not enumerable (e.g. for continuous x^k variables), EXPECTATION, REGRET (and even CONSENSUS and AMSAA) cannot be applied directly, while our method is still viable with no modification.

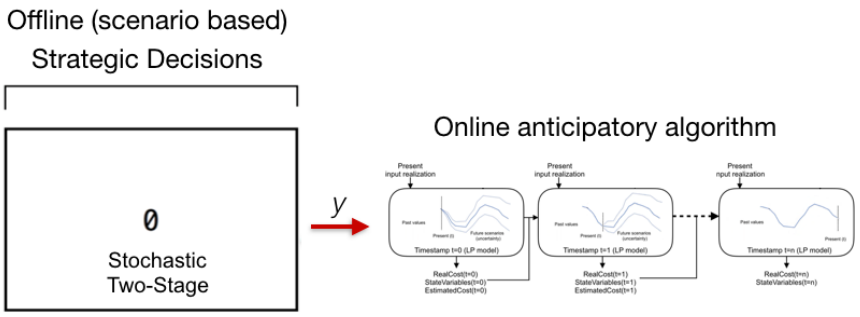

Figure 3.4: ANTICIPATE schema

5 TUNING

Our second technique for improving the online decision making consists in applying a parameter tuning phase to the greedy heuristic. In principle, this could be done by any suitable algorithm available from the literate, such as those from [LIDLC+16] or [HHLBS09]. However, we can take advantage of the convexity of **PH** to tackle the tuning problem in a principled fashion and obtain a guaranteed optimal parameters.

In particular, any decision made a stage k by the heuristic is a global optimum for **PH**. Now, convexity implies that any local minimum must be a global minimum. Local minima can be characterized in terms of the Karush-Kuhn-Tucker optimality conditions [Win04]. Essentially, *those conditions give us a set of constraints that must be satisfied by any solution that is compatible with the behavior of the greedy heuristic*. We can exploit this property to formulate the tuning problem as a mathematical program.

As a first step, we need to consider the form of the KKT conditions for

PH in a given scenario ω. Those are given by:

$$-\nabla_{x^k_\omega} f(..;\alpha_k) = \sum_{i=1}^{|e|} \lambda^k_{\omega,i} \nabla_{x^k_\omega} e_i + \sum_{i=1}^{|g|} \mu^k_{\omega,i} \nabla_{x^k_\omega} g_i \tag{3.9}$$

$$\mu^k_{\omega,i} g_i = 0 \qquad \forall i = 1..|g| \tag{3.10}$$

$$\mu^k_{\omega,i} \geq 0 \qquad \forall i = 1..|g| \tag{3.11}$$

$$\text{Eq. } (3.7, 3.8) \qquad \textit{– online problem constraints –}$$

where, for sake of readability, $f(y, x^k_\omega, s^k_\omega; \alpha_k)$ has been shortened to f, the i-th component (out of $|e|$) of $e(y, x^k_\omega, s^k_\omega)$ to e_i, and the i-th component (out of $|g|$) of $g(y, x^k_\omega, s^k_\omega)$ to g_i. The $\lambda^k_{\omega,i}$ and $\mu^k_{\omega,i}$ variables represent dual multipliers. Eq. (3.9) corresponds to the gradient cancellation condition, Eq. (3.10) to complementary slackness, Eq. (3.11) to dual feasibility ($\lambda^k_{\omega,i}$ is free), and Eq. (3.7), (3.8) to primal feasibility. Note that *here we use the heuristic cost function* f, rather than the "real" cost $\mathcal{F}$.

Then, we rely on the KKT conditions to define a model for an additional offline processing step, whose goal is to find the optimal values of the α_k parameters for a given set of scenarios. Such model is given by:

$$\min \frac{1}{|\Omega|} \sum_{\omega \in \Omega} \sum_{k=1}^{n} \mathcal{F}(y, x^k_\omega, s^k_\omega) \qquad (\mathbf{PT})$$

$$\begin{aligned}
&\text{s.t. Eq. } (3.2) - (3.4) && \textit{– flattened and state –} && \forall \omega \in \Omega \\
&\quad \text{Eq. } (3.5), (3.6) && \textit{– initial state –} && \\
&\quad \text{Eq. } (3.9) - (3.11) && \textit{– KKT cond –} && \forall \omega \in \Omega, \forall k = 1 \dots n
\end{aligned}$$

This is a stochastic two-stage model where the first stage variables are the α_k parameters (appearing in the equations for the KKT conditions), and the recourse actions (i.e. second stage variables) are the decisions x^k_ω that the heuristic would make in the considered scenarios – plus the related states s^k_ω and the $\lambda^k_{\omega,i}$ and $\mu^k_{\omega,i}$ multipliers. The problem goal is to minimize the expected cost over all stages and scenarios.

Solving **PT** yields an optimal parameter vector for the considered scenario set Ω. The offline decisions y are still made using **PO**, while the

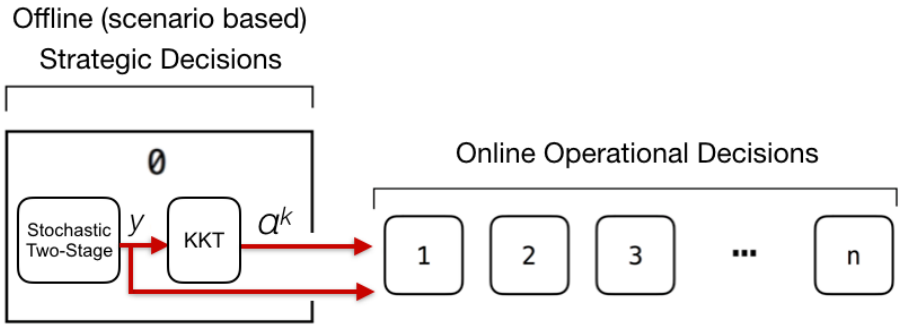

Figure 3.5: TUNING schema

online decisions are made via **PH**, *with the optimized parameters*. We refer to this method as TUNING. Intuitively, this approach should allow to retain some of the benefits of ANTICIPATE, without increasing the online computational cost. The price to pay is a considerably larger offline cost.

6 ACKNOWLEDGE

We now move to explore the second improvement direction: rather than trying to overcome the limitations of the online approach, we make the offline solver aware of the online heuristic.

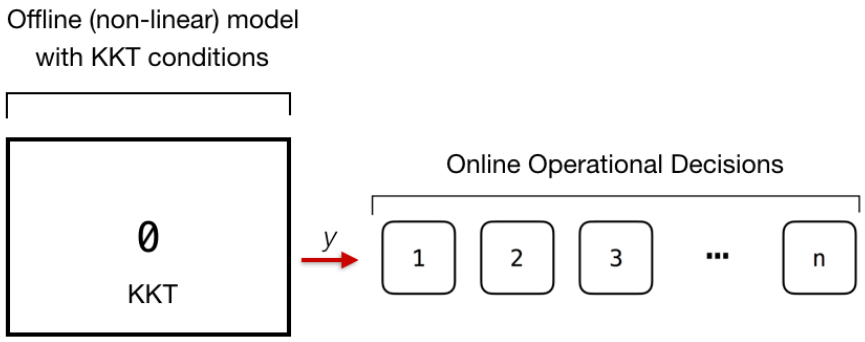

Figure 3.6: ACKNOWLEDGE schema

We achieve this by simply injecting the KKT conditions from Eq. (3.9) - (3.11) as constraints in **PO**. Similarly to what done to **PT**, this forces

all x_ω^k variables in the offline problem to take the values that would be actually assigned by the heuristic. Overall, we get the following problem:

$$\begin{aligned} &\min\ f_o(y) + \frac{1}{|\Omega|}\sum_{\omega\in\Omega}\sum_{k=1}^{n}\mathcal{F}(y, x_\omega^k, s_\omega^k) && (\mathbf{PACK}) \\ &\text{s.t. Eq. } (3.2)-(3.4) && \text{– } \textit{flattened and state} \text{ –} && \forall\omega\in\Omega \\ &\quad \text{Eq. } (3.5), (3.6) && \text{– } \textit{initial state} \text{ –} \\ &\quad \text{Eq. } (3.9)-(3.11) && \text{– } \textit{KKT cond} \text{ –} && \forall\omega\in\Omega, \forall k = 1\dots n \end{aligned}$$

Similarly to **PO**, this is a two-stage stochastic program. The first stage variables are the offline decisions y, while the recourse actions are x_ω^k – plus s_ω^k, $\lambda_{\omega,i}^k$, and $\mu_{\omega,i}^k$.

Once an offline decision vector has been found via **PACK**, the online decisions can be made via the original heuristics. We refer to this method as ACKNOWLEDGE. The method achieves integration at the cost of offline solution time, because of the additional variables in **PACK** and the presence of non-linearities in Eq. (3.10).

7 ACTIVE

Finally, we can combine these two methods to obtain the ACTIVE method that is composed by an offline part with KKT conditions that optimizes the offline decisions y and α^k (i.e. **PACT**).

$$\begin{aligned} &\min\ f_o(y) + \frac{1}{|\Omega|}\sum_{\omega\in\Omega}\sum_{k=1}^{n}\mathcal{F}(y, x_\omega^k, s_\omega^k) && (\mathbf{PACT}) \\ &\text{s.t. Eq. } (3.2)-(3.4) && \text{– } \textit{flattened and state} \text{ –} && \forall\omega\in\Omega \\ &\quad \text{Eq. } (3.5), (3.6) && \text{– } \textit{initial state} \text{ –} \\ &\quad \text{Eq. } (3.9)-(3.11) && \text{– } \textit{KKT cond} \text{ –} && \forall\omega\in\Omega, \forall k = 1\dots n \end{aligned}$$

The decision variables of **PACT** are, in this case, y, x_ω^k, s_ω^k, $\lambda_{\omega,i}^k$, $\mu_{\omega,i}^k$ and crucially α^k. The online decisions are then taken using the origi-

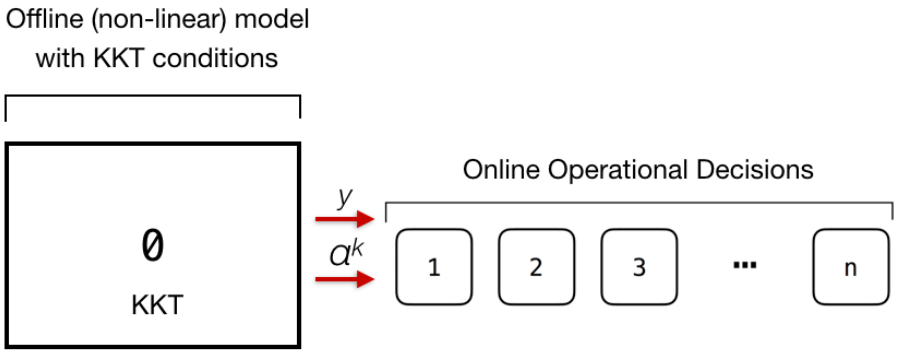

Figure 3.7: ACTIVE schema

nal heuristics, but its behavior will be affected also in this case by the "parameter schedule" $\alpha^1, \ldots \alpha^n$ produced by solving **PACT**. The difference in this case is that y and α^k are both optimized at the same time and by considering the KKT. The computational time will be higher but with, hopefully, better solution quality to steer online heuristic behavior.

8 Method Comparison

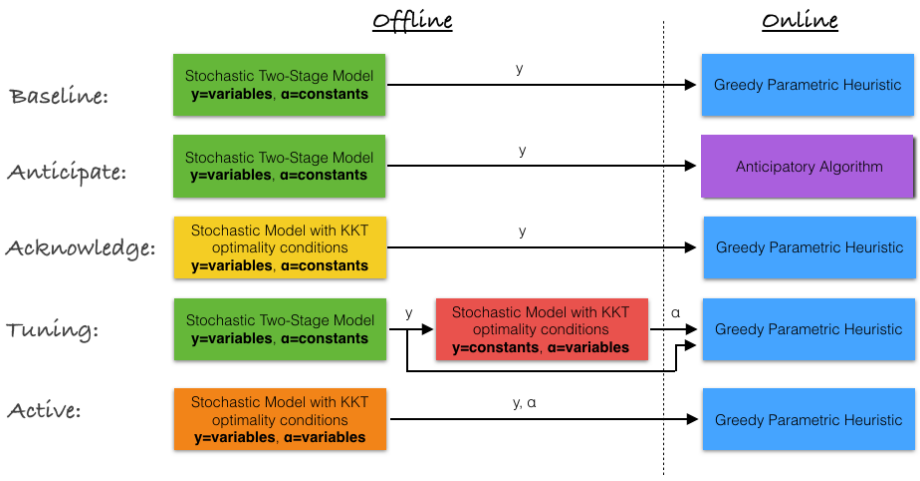

Figure 3.8: Proposed methods for Offline and Online integration

We start from the observation that many practical applications require to make interdependent offline and online decisions. The simplest and most common approach to tackle such problems is to deal with the offline and online phase separately, respectively (e.g.) via a sampling-based method and a heuristic: we consider these methods as components of our baseline. However, we will show that substantial improvements can be obtained by treating the two phases in an integrated fashion.

We propose four methods to improve the baseline in different directions, each altering either the offline or the online component of the solution process, so that the two play better together. Our methods are applicable provided that some simple but important assumptions are satisfied: 1) the uncertainty is exogenous; 2) in the baseline, a two-stage stochastic optimization model is used for the offline phase; 3) in the baseline, the online heuristic can be stated as convex optimization problem.

Selecting the suitable technique requires to consider the available time constraints for all offline and online decisions to use the most suitable method. We believe our techniques represent a significant step toward integrated offline/online optimization in complex systems.

Figure 3.8 summarizes the design of our methods, highlighting the techniques used in each phase, their decision variables and the output.

Chapter 4

Instantiating the Integrated Offline/Online Methods

In this section we present our case studies. The first one (an energy management system) was originally considered in [DFLMB17]: since it features continuous online decision variables, it is not amenable to existing approaches such as EXPECTATION or REGRET. The second use case (a Vehicle Routing Problem variant) is meant to provide a realistic, dramatically different, example of how the methods can be instantiated: it features discrete online decisions, and allows a quality comparison with classical algorithms because in such cases ANTICIPATE leads to the same results as EXPECTATION (although with different solution times). The case studies have been chosen to show that our methods work with both discrete and numerical decision variables.

1 Distributed Energy System: the Virtual Power Plant Case Study

The progressive shift towards decentralized generation in power distribution networks has made the problem of optimal Distributed Energy Resources (DERs) operation increasingly constrained. This is due

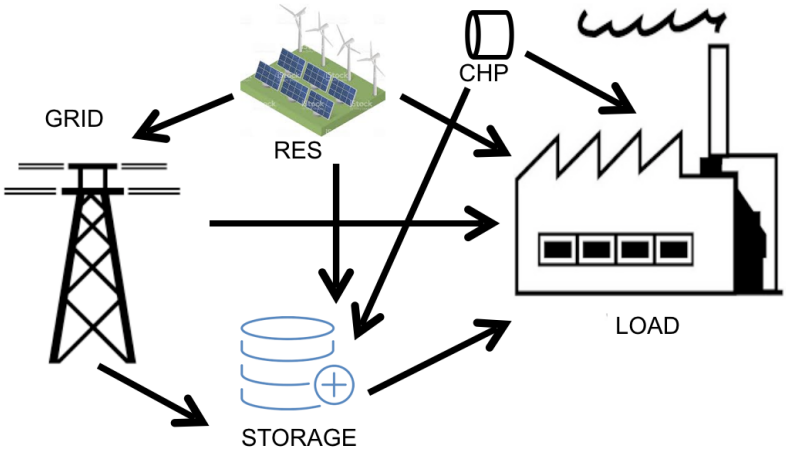

Figure 4.1: A typical Virtual Power Plant Energy Management System

to the integration of flexible (deterministic) energy systems with the strong penetration of (uncontrollable and stochastic) Renewable Energy Sources (RES). The integration of these resources into power system operation requires a major change in the current network control structure. This challenge can be met by using the Virtual Power Plant (VPP) concept, which is based on the idea of aggregating the capacity of many DERs, (i.e. generation, storage, or demand) to create a single operating profile and manage the uncertainty. A typical VPP is a large plant with high (partially shiftable) electric and thermal loads, renewable energy generators and electric and thermal storages (see Figure 4.1).

Making decisions under uncertainty pervades the planning and operation of energy systems and one of the most used assumption is that the distribution of future uncertainty is available for sampling, e.g. thanks to historical data and/or predictive models. In particular, the assumption that the distribution of future (exogenous) uncertainty is independent of current decisions is present in a variety of applications [HB09].

We consider a VPP Energy Management System (EMS) (see [MCM+13]) with partially shiftable loads, renewable energy generators, storage sys-

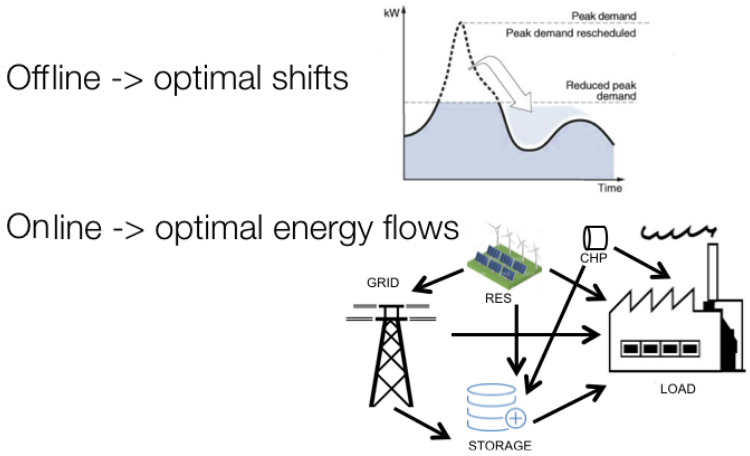

Figure 4.2: Offline/Online Decision Making in VPP

tems, and grid-connection. The goal is to decide the minimum-cost energy flows at each online stage (see [CBRJ15]). The uncertainty stems from uncontrollable deviations from the planned shifts and from the presence of Renewable Energy Sources (RES) (see [PBBA$^+$11a, BMRY15]). We assume that the RES production forecast is good enough that its error in each stage can be considered an independent random variable. Based on actual energy prices and on the availability of DERs, the EMS decides: 1) how much energy should be produced; 2) which generators should be used for the required energy; 3) whether the surplus energy should be stored or sold to the energy market; 4) the load shifts planned offline. Optimizing the use of energy can lead to significant economic benefits, and improve the efficiency and stability of the electric system (see e.g. [PBBA$^+$11a]).

Unlike in most of the existing literature, we acknowledge that in many practical cases, *the load shifts can be planned offline*, while the energy balance should be maintained online by managing energy flows among the grid, the renewable and traditional generators, and the storage systems. Intuitively, handling these two phases in an integrated fashion should lead to some benefits, thus making the VPP EMS a good candidate for grounding our approach (see Figure 4.2).

1.1 Instantiating the Baseline Model

The offline problem is modeled via Mixed Integer Programming (MILP) and it is given by:

$$\min \frac{1}{|\Omega|} \sum_{\omega \in \Omega} \sum_{g \in G} \sum_{k=1}^{n} c_g^k x_{g,\omega}^k \qquad (\mathbf{P1.1})$$

$$\text{s.t. } \tilde{L}_\omega^k = \sum_{g \in G} x_{g,\omega}^k \qquad \forall \omega \in \Omega, \forall k = 1, \dots n \tag{4.1}$$

$$\underline{x}_g \leq x_{g,\omega}^k \leq \overline{x}_g \qquad \forall \omega \in \Omega, \forall k = 1, \dots n \tag{4.2}$$

$$0 \leq \gamma_\omega^k \leq \Gamma \qquad \forall k = 1, \dots n \tag{4.3}$$

$$\gamma_\omega^{k+1} = \gamma_\omega^k + \eta x_{0,\omega}^k \quad \forall \omega \in \Omega, \forall k = 1, \dots n-1 \tag{4.4}$$

$$x_{1,\omega}^{k+1} = \hat{R}_k + \xi_{R,\omega}^k \qquad \forall \omega \in \Omega, \forall k = 1, \dots n \tag{4.5}$$

$$\tilde{L}_\omega^{k+1} = \hat{L}_k + y_k + \xi_{L,\omega}^k \quad \forall \omega \in \Omega, \forall k = 1, \dots n \tag{4.6}$$

$$\sum_{k=t}^{t+m} y_k = 0 \qquad \forall t = 1, \dots n-m \tag{4.7}$$

$$\underline{y}^k \leq y_k \leq \overline{y}^k \qquad \forall k = 1, \dots n \tag{4.8}$$

where Eq.(4.1)−(4.6) define the flattened problem, and Eq. (4.7)−(4.8) the feasible space for the offline variables y. In particular, Eq.(4.4) − (4.6) represent the transiction function, where $\hat{R}_k$ and $\hat{L}^k$ are the estimated RES production and load, and ξ_R^k and ξ_L^k are the corresponding errors (random variables). We assume that the errors follow roughly a Normal distribution $N(0, \sigma^2)$, and that the variance σ^2 is such that the 95% confidence interval corresponds to $\pm 20\%$ of the estimated value [GYI02]. The y^k variable represents the (offline planned) shift from the estimated load. Eq. (4.7) ensures that the shifts respect a local balance. The initial battery charge γ_ω^0 is identical for all scenarios.

Based on the shifts produced by the offline step, and adjusted to take into account the uncertainty, the online heuristic minimizes the operational cost and covers the energy demand by manipulating flows between nodes in $g \in G$. We assume the index 0 refers to the storage system and index 1 to the RES generators. The stages represent periods

long enough to treat the corresponding flow decisions as independent. The heuristic can be formulated as an LP model:

$$\min \sum_{k=1}^{n} \sum_{g \in G} c_g^k x_g^k \qquad (\mathbf{P1.2})$$

$$\text{s.t. } \tilde{L}^k = \sum_{g \in G} x_g^k \tag{4.9}$$

$$0 \leq \gamma_k + \eta x_0^k \leq \Gamma \tag{4.10}$$

$$\underline{x}_g \leq x_g^k \leq \overline{x}_g \tag{4.11}$$

where n is the number of online stages, and x_g^k represents the flow from g to the VPP (if positive) or in the reverse direction (if negative). All flows must respect the physical bounds $\underline{x}_g$ and $\overline{x}_g$. The flow costs c_g^k correspond to the problem parameters α^k in **PH**. The state variables are the RES energy flow x_1^k, the load to be satisfied $\tilde{L}^k$, and the battery charge γ^k. The battery upper limit is Γ and η is the charging/discharging efficiency.

1.2 Instantiating ANTICIPATE

A model for the ANTICIPATE approach can be obtained by applying in an almost straightforward fashion the definitions from Section 4:

$$\min \sum_{g \in G} c_g^h x_g^h + \frac{1}{|\Omega|} \sum_{\omega \in \Omega} \sum_{k=h+1}^{n} \sum_{g \in G} c_g^k x_{g,\omega}^k \qquad (\mathbf{P1.3})$$

$$\text{s.t. Eq. } (4.9) - (4.11) \qquad \textit{– online problem constraints –}$$

$$\text{Eq. } (4.1) - (4.3) \qquad \forall k > h \qquad \textit{– flattened –}$$

$$\text{Eq. } (4.4) - (4.6) \quad \forall k \geq h, \text{ with } s_\omega^h = s^h \text{ and } x_\omega^h = x^h \qquad \textit{– state –}$$

Note that **P1.3**, although potentially large, is a Linear Program and can be solved in polynomial time.

1.3 Instantiating TUNING

We start by formulating the KKT conditions for the online heuristic in a single scenario, thus obtaining:

$$-c_g^k = \lambda_\omega^k + \mu_{g,\omega}^k - \nu_{g,\omega}^k \qquad \forall g \in G \qquad (4.12)$$

$$\mu_{g,\omega}^k(x_{g,\omega}^k + \overline{x}_g) = 0 \qquad \forall g \in G \qquad (4.13)$$

$$\nu_{i,\omega}^k(\underline{x}_g - x_{g,\omega}^t) = 0 \qquad \forall g \in G \qquad (4.14)$$

$$\hat{\mu}_\omega^k(\eta x_{0,\omega}^k + \gamma^k - \Gamma) = 0 \qquad (4.15)$$

$$\hat{\nu}_\omega^k(\eta x_{0,\omega}^k + \gamma^k) = 0 \qquad (4.16)$$

$$\mu_{g,\omega}^k, \nu_{g,\omega}^k \geq 0 \qquad \forall g \in G \qquad (4.17)$$

$$\hat{\mu}_\omega^k, \hat{\nu}_\omega^k \geq 0 \qquad (4.18)$$

where $\mu_{g,\omega}^k$ and $\nu_{g,\omega}^k$ are the multipliers associated to the physical flow bounds, while $\hat{\mu}_\omega^k$ and $\hat{\nu}_\omega^k$ are associated to the battery capacity bounds. The multiplier λ_ω^k is associated to the balancing constraint, i.e. Eq. (4.9), and can be eliminated with a few algebraic transformations. Injecting the conditions in the offline model yields:

$$\min \frac{1}{|\Omega|} \sum_{\omega \in \Omega} \sum_{g \in G} \sum_{k=1}^{n} c_g^k x_{g,\omega}^k \qquad (\mathbf{P1.4})$$

$$\text{s.t. Eq. } (4.1) - (4.8) \qquad \textit{– offline problem –}$$

$$\text{Eq. } (4.26) - (4.32) \qquad \forall \omega \in \Omega, \forall k = 1, \ldots n \qquad \textit{– KKT cond –}$$

where the decision variables are y^k, $x_{g,\omega}^k$, $\mu_{g,\omega}^k$, $\nu_{g,\omega}^k$, $\hat{\mu}_\omega^k$, $\hat{\nu}_\omega^k$, based on the considered method. To those, we add the cost c_0^k associated to the flow between the VPP and the storage system (the only parameter that we allow the solver to adjust). Normally, there are neither economic penalties nor incentives for such flow, while there is a profit associated to flows from the VPP to the grid.

In particular, in TUNING we use **P1.2** to solve the y^k variables and then we use **P1.4** for c_0^k (i.e. our offline phase is divided in two parts).

1.4 Instantiating ACKNOWLEDGE

We formulate the KKT conditions for the online heuristic in a single scenario also in this method:

$$-c_g^k = \lambda_\omega^k + \mu_{g,\omega}^k - \nu_{g,\omega}^k \qquad \forall g \in G \qquad (4.19)$$

$$\mu_{g,\omega}^k(x_{g,\omega}^k + \overline{x}_g) = 0 \qquad \forall g \in G \qquad (4.20)$$

$$\nu_{i,\omega}^k(\underline{x}_g - x_{g,\omega}^t) = 0 \qquad \forall g \in G \qquad (4.21)$$

$$\hat{\mu}_\omega^k(\eta x_{0,\omega}^k + \gamma^k - \Gamma) = 0 \qquad (4.22)$$

$$\hat{\nu}_\omega^k(\eta x_{0,\omega}^k + \gamma^k) = 0 \qquad (4.23)$$

$$\mu_{g,\omega}^k, \nu_{g,\omega}^k \geq 0 \qquad \forall g \in G \qquad (4.24)$$

$$\hat{\mu}_\omega^k, \hat{\nu}_\omega^k \geq 0 \qquad (4.25)$$

where $\mu_{g,\omega}^k$ and $\nu_{g,\omega}^k$ are the multipliers associated to the physical flow bounds, while $\hat{\mu}_\omega^k$ and $\hat{\nu}_\omega^k$ are associated to the battery capacity bounds. The multiplier λ_ω^k is associated to the balancing constraint, i.e. Eq. (4.9), and can be eliminated with a few algebraic transformations. Injecting the conditions in the offline model yields:

$$\min \frac{1}{|\Omega|} \sum_{\omega \in \Omega} \sum_{g \in G} \sum_{k=1}^{n} c_g^k x_{g,\omega}^k \qquad (\mathbf{P1.4})$$

$$\text{s.t. Eq. } (4.1) - (4.8) \qquad \textit{– offline problem constraints –}$$

$$\text{Eq. } (4.26) - (4.32) \qquad \forall \omega \in \Omega, \forall k = 1, \dots n \qquad \textit{– KKT conditions –}$$

where the decision variables are y^k, $x_{g,\omega}^k$, $\mu_{g,\omega}^k$, $\nu_{g,\omega}^k$, $\hat{\mu}_\omega^k$, $\hat{\nu}_\omega^k$, based on the considered method. To those, we add the cost c_0^k associated to the flow between the VPP and the storage system (the only parameter that we allow the solver to adjust). Normally, there are neither economic penalties nor incentives for such flow, while there is a profit associated to flows from the VPP to the grid.

We recall that in ACKNOWLEDGE we consider y^k as decision variables and c_0^k as constant parameters using **P1.4** as offline phase.

1.5 Instantiating ACTIVE

We formulate the KKT conditions again for the online heuristic in a single scenario, thus obtaining:

$$-c_g^k = \lambda_\omega^k + \mu_{g,\omega}^k - \nu_{g,\omega}^k \qquad \forall g \in G \qquad (4.26)$$

$$\mu_{g,\omega}^k(x_{g,\omega}^k + \overline{x}_g) = 0 \qquad \forall g \in G \qquad (4.27)$$

$$\nu_{i,\omega}^k(\underline{x}_g - x_{g,\omega}^t) = 0 \qquad \forall g \in G \qquad (4.28)$$

$$\hat{\mu}_\omega^k(\eta x_{0,\omega}^k + \gamma^k - \Gamma) = 0 \qquad (4.29)$$

$$\hat{\nu}_\omega^k(\eta x_{0,\omega}^k + \gamma^k) = 0 \qquad (4.30)$$

$$\mu_{g,\omega}^k, \nu_{g,\omega}^k \geq 0 \qquad \forall g \in G \qquad (4.31)$$

$$\hat{\mu}_\omega^k, \hat{\nu}_\omega^k \geq 0 \qquad (4.32)$$

where $\mu_{g,\omega}^k$ and $\nu_{g,\omega}^k$ are the multipliers associated to the physical flow bounds, while $\hat{\mu}_\omega^k$ and $\hat{\nu}_\omega^k$ are associated to the battery capacity bounds. The multiplier λ_ω^k is associated to the balancing constraint, i.e. Eq. (4.9), and can be eliminated with a few algebraic transformations. Injecting the conditions in the offline model yields:

$$\min \frac{1}{|\Omega|} \sum_{\omega \in \Omega} \sum_{g \in G} \sum_{k=1}^{n} c_g^k x_{g,\omega}^k \qquad (\mathbf{P1.4})$$

$$\text{s.t. Eq. } (4.1) - (4.8) \qquad \textit{– offline problem constraints –}$$

$$\text{Eq. } (4.26) - (4.32) \qquad \forall \omega \in \Omega, \forall k = 1, \ldots n \qquad \textit{– KKT conditions –}$$

where the decision variables are y^k, $x_{g,\omega}^k$, $\mu_{g,\omega}^k$, $\nu_{g,\omega}^k$, $\hat{\mu}_\omega^k$, $\hat{\nu}_\omega^k$, based on the considered method. To those, we add the cost c_0^k associated to the flow between the VPP and the storage system (the only parameter that we allow the solver to adjust). Normally, there are neither economic penalties nor incentives for such flow, while there is a profit associated to flows from the VPP to the grid.

In ACTIVE we use **P1.4** with both y^k and c_0^k as decision variables of the *same offline phase*. As a side effect, the naive **P1.1** heuristic will always choose to sell the surplus energy. ACTIVE allows the offline solver to

associate a "virtual profit" to storing energy, which enables addressing the original limitation at no online computational cost.

2 Results for the VPP

We performed an experimentation to compare the solution quality and run times of our methods. As references for comparison we use the baseline approaches, plus an optimal solver operating under perfect information.

2.1 Experimental Setup

Our methods are evaluated over different uncertainty realizations, obtained by sampling the random variables for the loads and RES generation in the VPP model. We consider a sample of 100 realizations for six different instances of each problem. We then run each approach on each realization and measure the cost and run time. The scenarios in our models, conversely, are not sampled, but programmatically chosen: for the VPP we consider four "extreme" scenarios where (resp.) the load and the RES generation are at low/high values and the VPP problem has 24 online stages.

We solve our LPs and MILPs using Gurobi, while for the non-linear problems we use BARON via the GAMS modeling system on the Neos server for optimization. The time limit is 100 seconds. We use data from two public datasets to define problem instances for a residential [EO15] and industrial plant[1].

We use data from two public datasets to test our models on a residential plant[EO15] with only PV energy production for renewable sources and an industrial plant[2] with eolic and PV production. We modify these datasets to obtain use cases for testing our models (see Table 4.1):

[1] https://data.lab.fiware.org/dataset/

[2] Available at `https://data.lab.fiware.org/dataset/`

1. RB is the baseline residential dataset;
2. RR is the residential dataset with an increase of renewable (i.e. PV) production;
3. RP is dataset UC1 where the market prices are different for the sale/purchase of energy from/to the grid;
4. IB is the industrial dataset with also eolic renewable production;
5. in IR we increase the renewable production as in RR;
6. in IP we consider IB with different market prices as in RP.

Methodologies for the estimation of hourly global solar radiation have been proposed by many researchers and in this work, we consider as a prediction the average hourly global solar radiation from [SE07] and we use assumption for wind prediction from [HLM$^+$12]. We then assume that the prediction errors in each timestamp can be modeled again as random variables. Specifically, we assume normally distributed variables with a variance such that the 95% confidence interval corresponds to $\pm 10\%$ of the prediction value. We assume physical bounds on CHP due to its Electrical Capability based on real generation data[BMRY15, EO15]. The initial battery states and the efficiency values are based on real generation data [BMRY15, EO15] and we assume there are physical bounds for storage system based on real data [BMRY15, EO15].

Load demand	Baseline dataset	Renewable peak	Different market Prices
Residential	RB	RR	RP
Industrial	IB	IR	IP

Table 4.1: Different use cases

2.2 Discussion

In Tables 4.2 and 4.3 we show the average costs and run time over the 100 input realizations for each approach for the VPP use case.

Online times refer to the sum of the stages. The baseline model (being an LP) appears to be rather efficient in terms of computation time, but yields solutions of limited quality. The ANTICIPATE method comes much closer to the oracle solver, at the cost of a higher, but still reasonable, online run time. The ACTIVE method incurs substantially larger offline solution times, but it manages to beat or match the ANTICIPATE solution quality by making use of the original, straightforward, online heuristic. Table 4.3 shows a comparison among the computational times of all the proposed method to help understanding the potential of each method both in terms of offline and online computational cost.

	Daily Cost (k€)					
Instance	Oracle	Baseline	ANTICIPATE	ACKNOWLEDGE	TUNING	ACTIVE
RB	**331.36**	404.62	**342.06**	382.44	391.18	346.60
RR	**247.21**	311.14	**265.32**	297.77	294.75	266.80
RP	**393.81**	462.57	404.32	435.11	422.92	**408.72**
IB	**798.38**	923.24	822.24	894.33	883.99	**817.11**
IR	**565.60**	684.19	580.17	625.83	609.81	**577.93**
IP	**856.95**	984.90	**874.58**	950.81	901.27	888.76

Table 4.2: Cost values for the different VPP models

	Offline phase (sec)				**Online phase (sec)**	
Instance	Baseline	ACKNOWLEDGE	TUNING	ACTIVE	Heuristic	ANTICIPATE
RB	0.184	10.453	10.980	27.884	0.778	5.011
RR	0.190	9.996	9.473	31.992	0.772	5.017
RP	0.185	10.944	11.221	30.772	0.775	5.009
IB	0.346	15.437	13.466	38.913	0.839	5.430
IR	0.341	16.777	12.994	39.184	0.832	5.423
IP	0.348	15.768	15.443	37.777	0.835	5.420

Table 4.3: Computation time for the different VPP model stages

We show, for the VPP, the average values of each hourly optimized flow

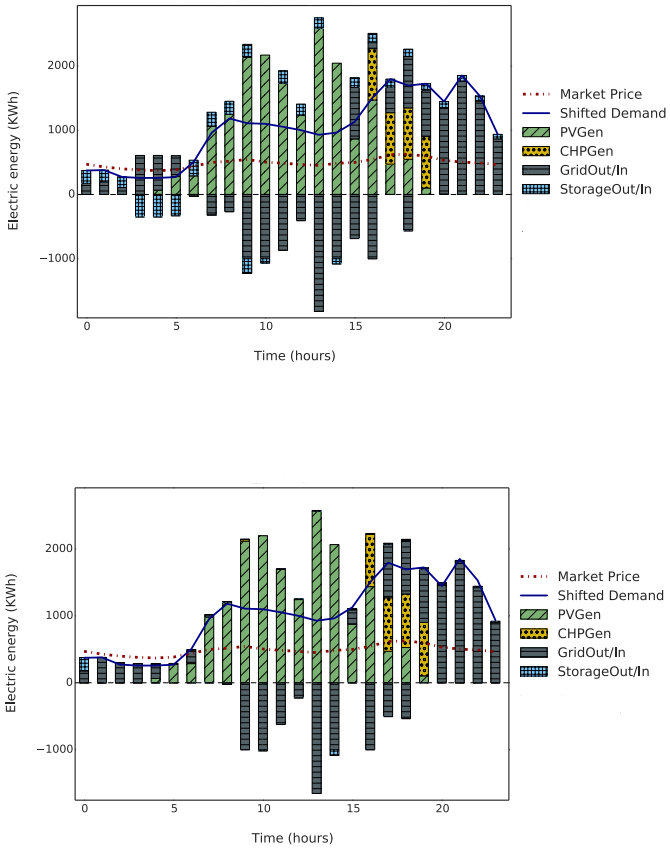

Figure 4.3: Oracle (up) and Baseline (down) optimal power flows for instance RR

over the 100 realizations for each proposed model in instance RR. We can see, in Fig. 4.3, the limits of using a non anticipatory algorithm, compared for example the Oracle optimization, since it is not possible to acquire energy from the grid in advance (i.e. when the cost is lower) and/or to sell energy to the grid in periods of highest price on the market or when more energy is available from renewable sources.

Moreover the exchange of energy with the storage system is almost never used, i.e. to store RES energy. In Fig. 4.5, it is possible to see that, near the peak of renewable energy production, the ACTIVE model

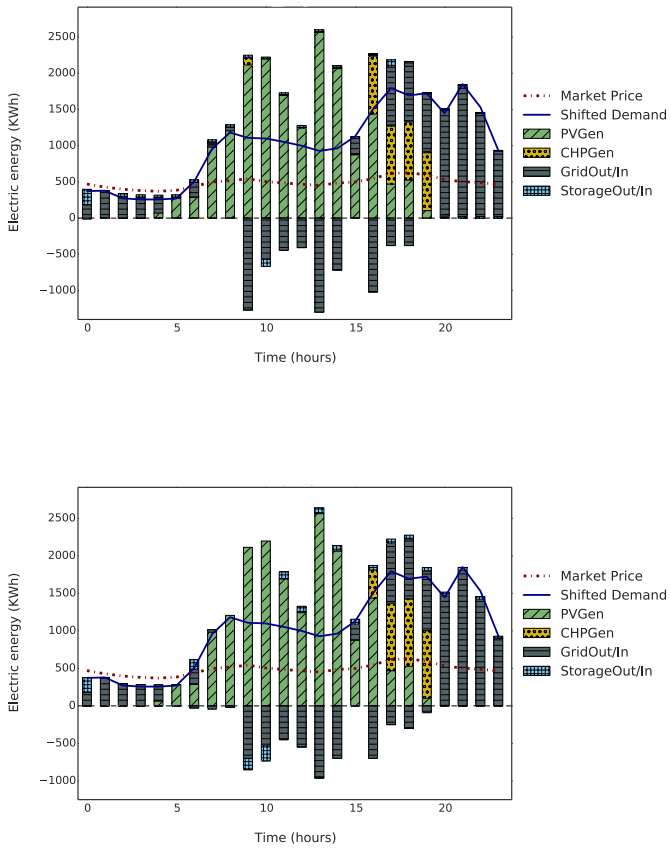

Figure 4.4: ACKNOWLEDGE (up) and TUNING (down) optimal power flows for instance RR

accumulates energy in the storage and uses in a more balanced way the energy present in the storage system compared to the baseline model represented in Fig. 4.3 and which never uses the storage system. Furthermore, still looking at Fig. 4.5, it can be seen that ANTICIPATE has peaks of energy sold on the network near the increase in electricity prices on the market. In Fig. 4.4 and Fig. 4.5 it is possible to notice the more consistent use of the storage system. We can see that, by optimizing the virtual storage cost in the offline phase, we can improve solution quality in term of cost (see Table 1) by using the storage sys-

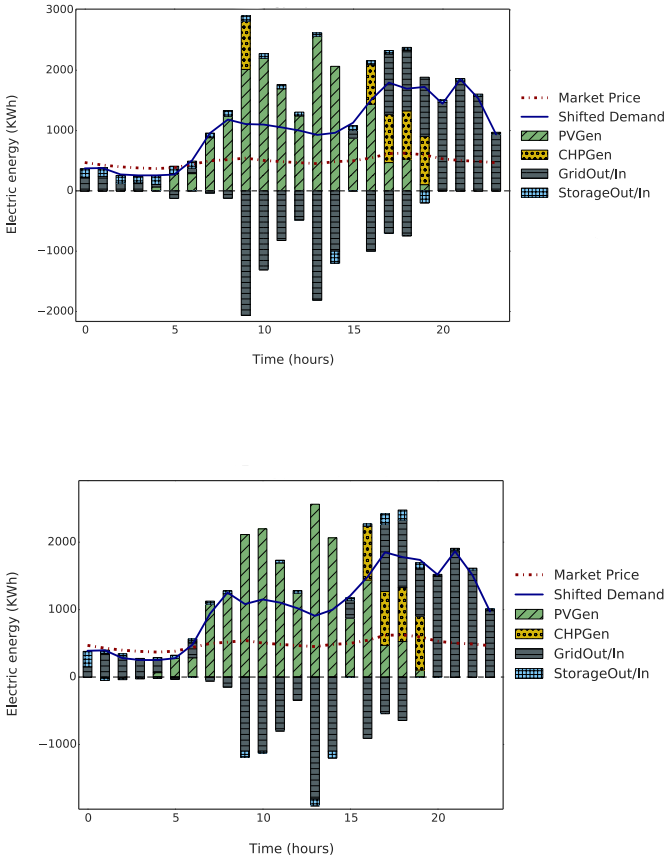

Figure 4.5: ANTICIPATE (up) and ACTIVE (down) optimal power flows for instance RR

tem. Since the online solver has the ability to sell energy on the market, and storing energy has no profit, it ends up in always selling unless the virtual cost is employed.

In Fig. 4.6 is shown the comparison between the optimized α in the two methods ACTIVE and TUNING. Moreover, we show also the market price trend to compare the trends of the two methods. It is interesting to notice how both the methods try to increase or decrease the virtual cost of the storage to promote the storage of energy in anticipation of future increases of energy price. In Fig. 4.7 it is possible to observe

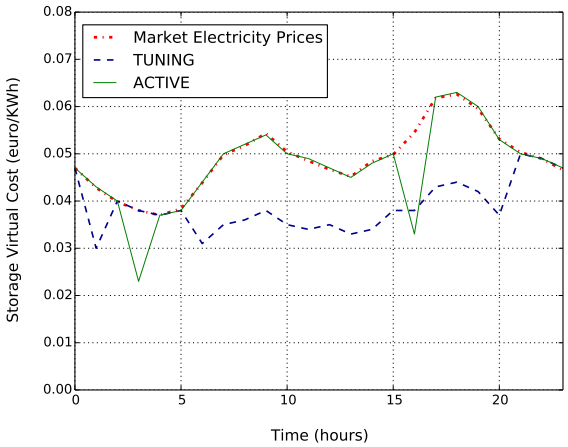

Figure 4.6: Market prices and optimized alpha in ACTIVE and TUNING for instance RB

the different use of the storage system due to the optimized parameters injected in the online solver to guide the heuristic decisions. Since the online (baseline) solver has the ability to sell energy on the market and storing energy has no profit, it ends up in always selling unless the virtual cost is employed (ACTIVE).

3 The Vehicle Routing Problem Case Study

We consider a variant of the Capacitated VRP with uncertain travel times (see [TV02, BSL96, LLP12, TDVWDK13]). The problem consists in establishing the paths of a set of vehicles to serve a set of customers. All vehicles have a finite capacity, and customers have a known demand and can be visited by a single vehicle. There are n fully connected customers/nodes, with node 0 being the (single) depot.

Customer assignments must be done offline, while the vehicle routes are chosen online. We assume that, whenever a node is reached, its binary "state" becomes known, and with that the (uniform) distributions followed by the travel times of all its outgoing arcs.

Formally, this results in bi-modally distributed, statistically dependent,

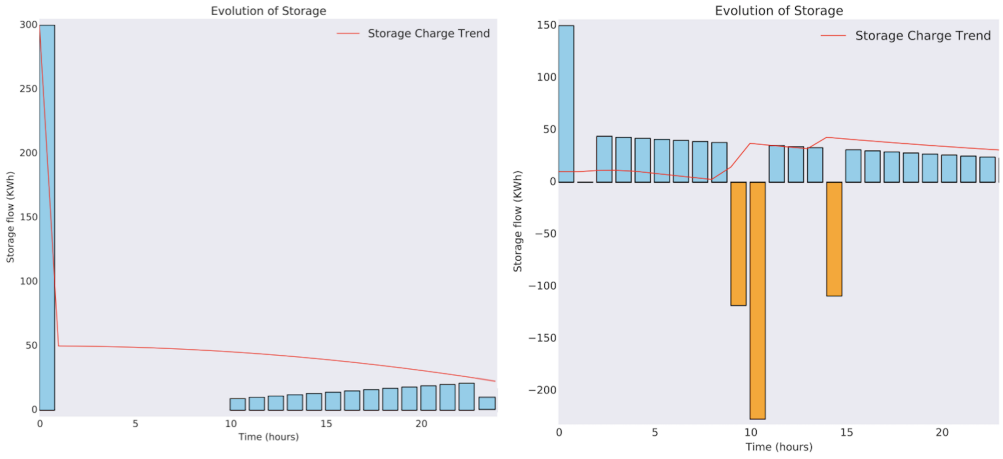

Figure 4.7: Baseline (left) and ACTIVE (right) online storage flow for instance RB

travel times. The objective is to minimize the total travel time.

3.1 Instantiating the Baseline Model

The online heuristic consists in simply picking the outgoing arc with the shortest travel time. This can be modeled also as a simple Integer Program. Let h be the current node, then we have:

$$\min \sum_{j \in V_h} c_{hj} x_{hj} \qquad (\mathbf{P2.1})$$

$$\text{s.t.} \sum_{j \in V_h} x_{h,j} = 1 \qquad (4.33)$$

$$x_{h,j} \in \{0,1\} \qquad \forall j \in V \qquad (4.34)$$

where $x_{hj} = 1$ iff we choose to move from h to j, V_h is the set of nodes that still needs to be visited (and it always include the depot), and the travel times c_{hj} are the heuristic parameters. **P2.1** does not apparently satisfy our assumptions, due to the integer variables. However, *its LP relaxation has always an integer solution, banning degenerate cases* (i.e. arcs with the same cost). We can therefore relax the integrality requirement without loss of generality. The transition function is given

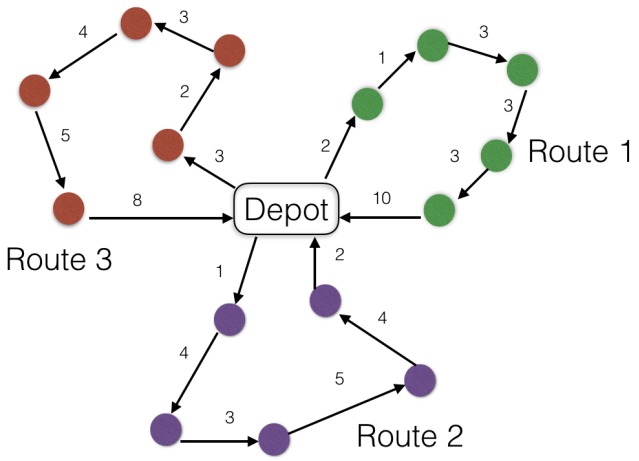

Figure 4.8: An Example of Vehicle Routing Problem

by:

$$V_{h^*} = V_h \setminus \{h^*\} \tag{4.35}$$

$$c_{h^*,j} = \xi_{h^*,j} \tag{4.36}$$

where h^* is the index of the next node selected by the heuristic and $\xi_{h^*,j}$ is the travel time from h^* to j (a random variable). Note also that in this case *the index of the online stage is implicitly given by* h. We take advantage of this and reduce the notation clutter by moving the ω index to apex position.

We tackle the offline problem via Mixed Integer Linear Programming, which forbids us to directly embed the non-linear Eq. (4.35) in the model. In practice, however, the equation states that 1) each vehicle should serve only its assigned customers, and 2) the visit should form a single loop. Both are well known VRP constraints and can be lin-

earized. In particular, we use the model:

$$\min \frac{1}{|\Omega|} \sum_{\omega \in \Omega} \sum_{k \in K} \sum_{i,j \in V} \xi_{i,j}^{\omega} x_{k,i,j}^{\omega} \qquad (\mathbf{P2.2})$$

$$\text{s.t.} \sum_{j \in V} x_{k,i,j}^{\omega} = y_{k,i} \qquad \forall k \in K, \forall i \in V \qquad (4.37)$$

$$\sum_{i \in V} x_{k,i,j}^{\omega} = y_{k,j} \qquad \forall k \in K, \forall j \in V \qquad (4.38)$$

$$y_{k,0} = 1 \qquad \forall k \in K \qquad (4.39)$$

$$t_{k,j}^{\omega} \geq t_{k,i}^{\omega} - M + (M+1) x_{k,i,j}^{\omega} \qquad \begin{matrix} \forall k \in K, \\ \forall i,j \in V, V^{+} \end{matrix} \qquad (4.40)$$

$$t_{k,0}^{\omega} = 0 \qquad \forall k \in K \qquad (4.41)$$

$$\sum_{i \in V} q_i y_{k,i} \leq C_k \qquad \forall k \in K \qquad (4.42)$$

$$\sum_{k \in K} y_{k,i} = 1 \qquad \forall i \in V^{+} \qquad (4.43)$$

where all constraints where an ω apex appears should be posted $\forall \omega \in \Omega$. All x and y variables are binary, and $y_{ki} = 1$ iff customer i should be visited by vehicle k. We have $M = |V|$, and $V^{+} = V \setminus \{0\}$. Eq.(4.37)$-$(4.41) define the flattened problem, and Eq. (4.42) $-$ (4.43) define the feasible space of the offline decision variables. For sake of simplicity, we eliminate subloops by keeping track of the visiting order t_{ki}^{ω} of each node for each vehicle: this is a simple, but not particularly effective method, because it relies on big-Ms and reduces the quality of the LP bound [MTZ60].

3.2 Instantiating ANTICIPATE

The ANTICIPATE method can be instantiated for each vehicle k separately, by first restricting the focus to the set of nodes V_h, and then by applying the definition from Section 4 and linearizing Eq. (4.35) in the

baseline offline problem, we get:

$$\min \sum_{j \in V_h} c_{h,j} x_{k,i,j} + \frac{1}{|\Omega|} \sum_{\omega \in \Omega} \sum_{i,j \in V_h} \xi^{\omega}_{i,j} x^{\omega}_{k,i,j} \qquad (\mathbf{P2.3})$$

s.t. Eq. (4.33) *– online problem constraints –*

Eq. (4.37) restricted to $V_h \setminus \{0\}$ (4.44)

Eq. (4.38) $-$ (4.40) restricted to V_h *– state transition –*

$t^{\omega}_{k,h} = 0$ (4.45)

where Eq. (4.45) means that the vehicle path should start from the current node h (and end as usual in the depot).

3.3 Instantiating TUNING

The first step is formulating the KKT conditions for **P2.1**. In this case after some algebraic transformations, for a given vehicle k, node h, and scenario ω we obtain:

$$(c_{hj} + \lambda^{\omega}_{k,h}) x^{\omega}_{k,h,j} = 0 \qquad \forall j \in V_h \qquad (4.46)$$

$$(c_{hj} + \lambda^{\omega}_{k,h}) \geq 0 \qquad \forall j \in V_h \qquad (4.47)$$

where $\lambda^{\omega}_{k,h}$ is the multiplier for Eq. (4.33), and all other multipliers have been eliminated. The main difficulty is again dealing with the set V_h, which is part of the state and should be constructed dynamically in the offline problem. Here, we handle V_h by introducing fresh variables r^{ω}_{kji} such that $r^{\omega}_{kji} = 1$ iff node i has been visited when node j is reached. The semantic is enforced via additional non-linear constraints in the

offline model. The latter is given by:

$$\min \frac{1}{|\Omega|} \sum_{\omega \in \Omega} \sum_{k \in K} \sum_{i,j \in V} \xi^{\omega}_{i,j} x^{\omega}_{k,i,j} \qquad (\mathbf{P2.4})$$

$$\text{s.t. Eq. (4.37)} - \text{(4.43)} \qquad \textit{– offline problem constraints –}$$

$$(c_{ij} + \lambda^{\omega}_{k,i}) x^{\omega}_{k,i,j} (1 - r_{kji}) = 0 \quad \forall k \in K, \forall i,j \in V$$

$$(c_{ij} + \lambda^{\omega}_{k,i})(1 - r_{kji}) \geq 0 \quad \forall k \in K, \forall i,j \in V$$

$$r^{\omega}_{k,i,i} = y_{k,i} \quad \forall i \in V$$

$$r^{\omega}_{k,j,i} = r^{\omega}_{k,h,i} x^{\omega}_{k,h,j} \quad \forall i \in V, \forall h \in V, \forall j \in V$$

$$\underline{c}_{ij} \leq c_{ij} \leq \overline{c}_{ij} \quad \forall i,j \in V$$

The decision variables are y_{ki}, x^{ω}_{kij}, λ^{ω}_{ki}, r^{ω}_{kji}, plus the "virtual travel times" c_{ij}, i.e. the parameters for the online heuristic, always based on the method considered. The constraints on the r^{ω}_{kji} variables enforce the transitive property on the set of visited nodes. Bounding the virtual travel times is necessary to prevent the solver from building degenerate parameterizations for **P2.1** on purpose, which would trivially satisfy all KKT constraints and make the approach boil down to the baseline offline solver.

Also here, we recall that in TUNING we use **P2.2** to solve the y_{ki} variables and then we use **P2.4** for c_{ij} (i.e. our offline phase is divised in two parts).

3.4 Instantiating ACKNOWLEDGE

As usual, the first step is formulating the KKT conditions for **P2.1**. In this case after some algebraic transformations, for a given vehicle k, node h, and scenario ω we obtain:

$$(c_{hj} + \lambda^{\omega}_{k,h}) x^{\omega}_{k,h,j} = 0 \quad \forall j \in V_h \qquad (4.48)$$

$$(c_{hj} + \lambda^{\omega}_{k,h}) \geq 0 \quad \forall j \in V_h \qquad (4.49)$$

where $\lambda^{\omega}_{k,h}$ is the multiplier for Eq. (4.33), and all other multipliers have been eliminated. The main difficulty is again dealing with the set V_h,

which is part of the state and should be constructed dynamically in the offline problem. Here, we handle V_h by introducing fresh variables r^{ω}_{kji} such that $r^{\omega}_{kji} = 1$ iff node i has been visited when node j is reached. The semantic is enforced via additional non-linear constraints in the offline model. The latter is given by:

$$\min \frac{1}{|\Omega|} \sum_{\omega \in \Omega} \sum_{k \in K} \sum_{i,j \in V} \xi^{\omega}_{i,j} x^{\omega}_{k,i,j} \qquad (\mathbf{P2.4})$$

$$\text{s.t. Eq. (4.37)} - \text{(4.43)} \qquad \textit{– offline problem constraints –}$$

$$(c_{ij} + \lambda^{\omega}_{k,i}) x^{\omega}_{k,i,j}(1 - r_{kji}) = 0 \quad \forall k \in K, \forall i,j \in V$$

$$(c_{ij} + \lambda^{\omega}_{k,i})(1 - r_{kji}) \geq 0 \quad \forall k \in K, \forall i,j \in V$$

$$r^{\omega}_{k,i,i} = y_{k,i} \quad \forall i \in V$$

$$r^{\omega}_{k,j,i} = r^{\omega}_{k,h,i} x^{\omega}_{k,h,j} \quad \forall i \in V, \forall h \in V, \forall j \in V$$

$$\underline{c}_{ij} \leq c_{ij} \leq \overline{c}_{ij} \quad \forall i,j \in V$$

The decision variables are y_{ki}, x^{ω}_{kij}, λ^{ω}_{ki}, r^{ω}_{kji}, plus the "virtual travel times" c_{ij}, i.e. the parameters for the online heuristic, always based on the method considered. The constraints on the r^{ω}_{kji} variables enforce the transitive property on the set of visited nodes. Bounding the virtual travel times is necessary to prevent the solver from building degenerate parameterizations for **P2.1** on purpose, which would trivially satisfy all KKT constraints and make the approach boil down to the baseline offline solver.

Also here, we recall that in ACKNOWLEDGE we consider y_{ki} as decision variables and c_{ij} as parameters using **P2.4** as offline phase.

3.5 Instantiating ACTIVE

The first step in the ACTIVE is still formulating the KKT conditions for **P2.1**. In this case after some algebraic transformations, for a given vehicle k, node h, and scenario ω we obtain:

$$(c_{hj} + \lambda^{\omega}_{k,h}) x^{\omega}_{k,h,j} = 0 \quad \forall j \in V_h \qquad (4.50)$$

$$(c_{hj} + \lambda^{\omega}_{k,h}) \geq 0 \quad \forall j \in V_h \qquad (4.51)$$

where $\lambda^{\omega}_{k,h}$ is the multiplier for Eq. (4.33), and all other multipliers have been eliminated. The main difficulty is again dealing with the set V_h, which is part of the state and should be constructed dynamically in the offline problem. Here, we handle V_h by introducing fresh variables r^{ω}_{kji} such that $r^{\omega}_{kji} = 1$ iff node i has been visited when node j is reached. The semantic is enforced via additional non-linear constraints in the offline model. The latter is given by:

$$\min \frac{1}{|\Omega|} \sum_{\omega \in \Omega} \sum_{k \in K} \sum_{i,j \in V} \xi^{\omega}_{i,j} x^{\omega}_{k,i,j} \qquad (\mathbf{P2.4})$$

$$\text{s.t. Eq. (4.37)} - \text{(4.43)} \qquad \textit{– offline problem constraints –}$$

$$(c_{ij} + \lambda^{\omega}_{k,i}) x^{\omega}_{k,i,j} (1 - r_{kji}) = 0 \quad \forall k \in K, \forall i,j \in V$$

$$(c_{ij} + \lambda^{\omega}_{k,i})(1 - r_{kji}) \geq 0 \quad \forall k \in K, \forall i,j \in V$$

$$r^{\omega}_{k,i,i} = y_{k,i} \quad \forall i \in V$$

$$r^{\omega}_{k,j,i} = r^{\omega}_{k,h,i} x^{\omega}_{k,h,j} \quad \forall i \in V, \forall h \in V, \forall j \in V$$

$$\underline{c}_{ij} \leq c_{ij} \leq \overline{c}_{ij} \quad \forall i,j \in V$$

The decision variables are y_{ki}, x^{ω}_{kij}, λ^{ω}_{ki}, r^{ω}_{kji}, plus the "virtual travel times" c_{ij}, i.e. the parameters for the online heuristic, always based on the method considered. The constraints on the r^{ω}_{kji} variables enforce the transitive property on the set of visited nodes. Bounding the virtual travel times is necessary to prevent the solver from building degenerate parameterizations for **P2.1** on purpose, which would trivially satisfy all KKT constraints and make the approach boil down to the baseline offline solver.

Also here, we recall that in ACTIVE we use **P2.4** with both y_{ki} and c_{ij} as decision variables of the *same offline phase*.

4 Results for the VRP

4.1 Experimental Setup

Our methods are evaluated over different uncertainty realizations, obtained by sampling the random variables for the travel times in the VRP model. We consider a sample of 100 realizations for six different instances of each problem. We then run each approach on each realization and measure the cost and run time. The scenarios in our models, conversely, are not sampled, but programmatically chosen: for the VRP, each scenario corresponds to the mean travel times in one mode of the distribution. In the VRP the number depends on how many customers are assigned to each vehicle.
We solve our LPs and MILPs using Gurobi, while for the non-linear problems we use BARON via the GAMS modeling system on the Neos server for optimization. The time limit is 500 seconds for the VRP and we use modified version of classical instances[3], by including problems from 10 to 30 customers with one depot and different numbers of vehicles.

4.2 Discussion

Tables 4.4 and 4.5 report the same results in terms of costs and computation time for the VRP. Here the online times are summed over all the vehicles. The original online heuristic is very efficient, but coupled with the baseline offline model it does not come close to the oracle quality. The offline model (a Mixed Integer Linear Program) takes also considerably more time to be solved. ANTICIPATE, which in this case yields the same results as EXPECTATION with no time limit, yields substantially better solutions, but, being also MILP-based, it takes non-negligible time during the online phase. The ACTIVE results follow the same trend as the VPP: the solution quality matches or beats that of

[3] http://myweb.uiowa.edu/bthoa/TSPTWBenchmarkDataSets.htm

ANTICIPATE, at the cost of a higher offline computation time, though the gap wrt the baseline is now much smaller.

Instance	Total travel Time (t)					
	Oracle	Baseline	ANTICIPATE	ACKNOWLEDGE	TUNING	ACTIVE
I1	**146.10**	165.83	**151.23**	162.88	158.01	148.84
I2	**278.37**	347.28	299.67	320.43	312.33	**295.43**
I3	**372.82**	561.66	**477.16**	530.43	522.32	507.80
I4	**321.57**	381.45	342.94	368.94	355.32	**340.85**
I5	**503.65**	670.86	559.22	659.22	632.33	**543.92**
I6	**448.53**	871.87	**470.99**	605.88	584.33	504.82

Table 4.4: Travel time (cost) values for the different VRP models

Instance	Offline phase (sec)				Online phase (sec)	
	Baseline	ACKNOWLEDGE	TUNING	ACTIVE	Heuristic	ANTICIPATE
I1	1.699	3.442	4.356	6.255	0.255	7.134
I2	2.477	10.229	12.377	17.445	0.169	15.222
I3	2.532	15.999	19.323	25.938	0.554	18.024
I4	186.798	288.344	295.247	338.998	3.444	255.932
I5	243.330	300.232	312.222	357.543	5.248	313.656
I6	361.537	405.233	422.300	490.856	5.342	416.645

Table 4.5: Computation time for the different VRP model parts

From Fig. 4.9 to Fig. 4.10 we show the average online routing decisions over the 100 realizations for the same instance I2 (10 customers, one depot and two vehicles). The heatmaps shown below represent different colors for each vehicle and different color intensity for the number of times that each route has been chosen over the 100 realizations. We remember that our models make first offline decisions (i.e. assignment of clients for each vehicle) and then make routing (online) decisions. We remember also that in ACTIVE model we can have different offline decisions (compared to those made from the other three models) since

we inject KKT conditions in the offline part. We therefore propose an instance as example where the offline decisions are the same for all the models with the aim to observe the different online routing decisions. We show the most representative and interesting results for method comparison (i.e. Oracle-Baseline, ANTICIPATE-ACTIVE). Indeed, we have different trends with the same offline decisions. In particular, the Baseline model makes different routing decisions compared to the Oracle decisions: routes 2 -> 10, 3 -> 2, 6 -> 5 are never considered in the Baseline decisions while they are (with a certain probability) considered in the ANTICIPATE decisions. We can also notice that, the Baseline and ANTICIPATE models assume a (low) probability also for different routing decisions of vehicle 0 and this is not present in ACTIVE routing decisions. The ACTIVE routing decisions are equals to the Oracle ones for vehicle 0 in terms of probability and, for the other vehicle, they present routes (with the relative probability) never used by ANTICIPATE (e.g. 0 -> 6, 5 -> 7). Moreover, we can notice that ANTICIPATE presents online decisions with a higher probability but, in general, different from the most frequent decisions of the Oracle. Instead, ACTIVE makes more decisions with lower probability than ANTICIPATE, but considering more often decisions similar to the Oracle.

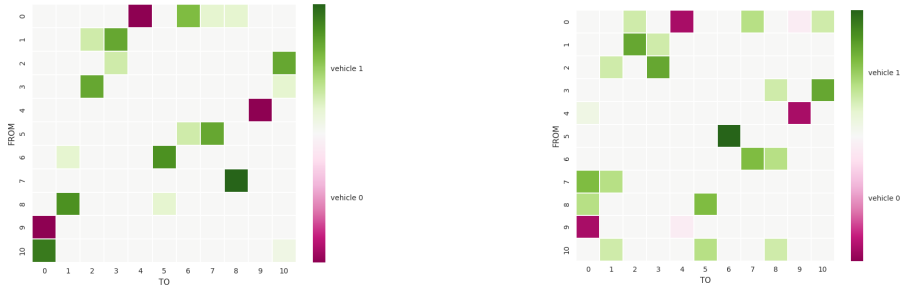

Figure 4.9: (left) Oracle and (right) Baseline routing decisions for instance I2

We proposed four alternative approaches based on the idea of making the offline and online solvers operate synergistically. All the techniques yield substantially improved solutions: ANTICIPATE matches the quality level of EXPECTATION, but it is applicable under more general as-

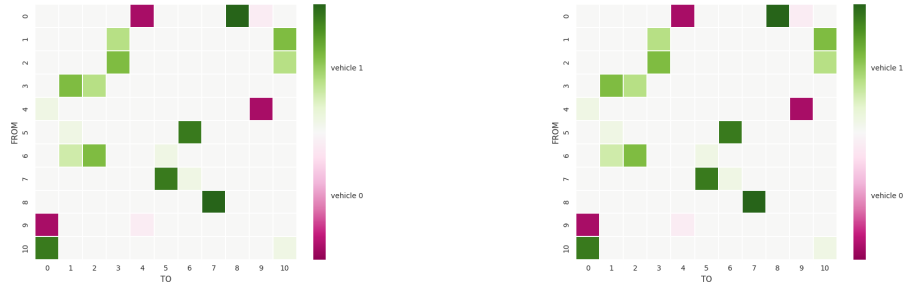

Figure 4.10: (left) ANTICIPATE and (right) ACTIVE routing decisions for instance I2

sumptions. ACTIVE often manages to beat ANTICIPATE (and therefore EXPECTATION) in terms of solution quality. While this comes at the price of a substantially increased offline computation time, the method achieves these results by using naive and very efficient online heuristics.

In the following part of the thesis we focus on how to manage the cost/quality trade-off of sampling-based anticipatory algorithms and we present three methods that can be applied to a generic anticipatory algorithm to reduce its online computational effort by exploiting offline information.

Chapter 5

Managing Cost-Quality Trade-Offs of Online Anticipatory Algorithms

1 Introduction

Optimization problems under uncertainty often benefit from making all or part of their decisions online, reacting and adapting to external events. In this context, stochastic online anticipatory algorithms have proved particularly effective (see e.g. [HB09]). However, many of such algorithms have a considerable computational cost, which may be problematic if (as it is often the case) online decisions must be taken within a short time frame.

In most practical settings, however, a substantial amount of time and information is available before the online problem is solved, in an *offline phase*. For example, one may have access to energy production forecasts, historical travel times in routing problems, results from test runs in cyber-physical systems. We refer to this sort of data as *offline information*. Usually, it is employed to characterize the uncertain elements and for sampling likely outcomes (i.e. scenarios). We will show how to exploit this information at a much deeper level.

We propose three hybrid offline/online methods that build over a given, sampling-based, anticipatory algorithm, and allow to match its solution quality at a fraction of the online computational cost. One of them can even rely on a deterministic algorithm, thus providing state-of-the art performance in problems for which no anticipatory approach is available. All our methods work by shifting part of the computation to the offline phase, where time limits are more relaxed and the costs can be better amortized (e.g. via parallelization).

We obtain our methods by combining three basic contributions: 1) a technique to estimate the probability of future outcomes, given past observations; 2) a scheme for building a "contingency table", with precomputed solutions to guide the online choices; and 3) an efficient fixing heuristic for adapting the precomputed solutions to run-time conditions.

We ground our approaches on a (numeric) energy management problem with uncertain loads and generation from Renewable Energy Sources (RES), and on a (discrete) Traveling Salesman Problem with uncertain travel times. We show how our methods reach a solution quality comparable with the anticipatory algorithm, with lower (or dramatically lower) online computational cost.

2 Motivations of "Taming" an Online Anticipatory Algorithm

Our goal is reducing the online computational cost of a given sampling-based anticipatory algorithm, referred to as A, by exploiting the existence of an offline phase. Such A algorithm is the main input for all our methods.

Similarly to [HB09], we view online optimization under uncertainty as a stochastic n-stage problem. At each stage some uncertainty is resolved, and some decision must be made. A stage k is associated to a

decision variable x_k (e.g. the power flows between loads and generators) and a state variable s_k (summarizing the effect of past decisions). All variables may be vector-valued.
We assume that uncertainty is *exogenous*, i.e. not affected by the decisions (e.g. the RES generation does not depend on how we choose to route it), and modeled via a set of random variables ξ_i. Which variables are observed at each stage depends on the state, and is controlled by a *peek* function:

$$O = peek(s_k) \tag{5.1}$$

which returns a set O with the indices of the observed variables. We will use the notation ξ_O to denote the observed ξ variables, and $\xi_{\bar{O}}$ for the unobserved ones.

2.1 Offline Information Availability

Defining a representative set of scenarios Ω is critical for the approach effectiveness and it is usually done by exploiting the available *offline information*. Here, we assume that the such offline information is a collection of observed uncertain values. This definition captures many practical cases (e.g. forecasts or predictions, historical data, data from test runs). More importantly, this means that *the offline information is in fact a collection of scenarios*. We will denote the offline information as I, index its element with ω, and assume (as it is usual) that I is representative of the true probability distribution of the random variables. A set Ω of scenarios for ANTICIPATE can be obtained by sampling a number of elements uniformly at random from I.

3 Building Block Techniques

All our methods rely on three techniques, which will be described in this section.

3.1 Probability Estimation for Scenario Sampling

Using a fixed set of scenarios (as in ANTICIPATE) is beneficial when the ξ_i variables are statistically independent. When they are not, however, the set of scenarios may loose relevance as uncertainty is resolved. For example, a scenario based on a cloudy day forecast becomes less likely if fair weather is observed at the beginning of online execution.

Formally, at stage k we wish to sample scenarios that are likely to occur given the past observations, i.e. to sample the unobserved variables $\xi_{\bar{O}}$ according to the conditional distribution $P(\xi_{\bar{O}} \mid \xi_O)$. If we draw the scenarios from the offline information (which guarantees physically meaningfulness), then sampling requires to estimate the conditional probabilities *of the elements in* I. From basic probability theory, this is given by the ratio of two joint probabilities:

$$\forall \omega \in I, \quad P(\xi^\omega_{\bar{O}} \mid \xi_O) = \frac{P(\xi_O \xi^\omega_{\bar{O}})}{P(\xi_O)} \tag{5.2}$$

where $P(\xi_O \xi^\omega_{\bar{O}})$ is the probability that observed values occur together with the remaining predictions from the scenario, and $P(\xi_O)$ is the probability that the values are observed. The joint probability at the numerator can be approximated via any density estimation method, such as Kernel Density Estimation [Sil18], Gaussian Mixture Models [GL94], or recent Deep Learning techniques such as Normalizing Flows [RM15] and Real NVP [DSDB16]. *Any such method can be trained on the offline information to obtain an estimator* $\tilde{P}(\xi)$ *for the joint distribution of the random variables.*

An estimator for the distribution $P(\xi_O)$ at the denominator can then be derived from $\tilde{P}(\xi)$ via marginalization, i.e. by averaging the contribution of the unobserved variables. We perform this step over all possible completions of the observed values *in the offline information*. Overall, we have:

$$\forall \omega \in I, \quad \tilde{P}(\xi^\omega_{\bar{O}} \mid \xi_O) = \frac{\tilde{P}(\xi_O \xi^\omega_{\bar{O}})}{\sum_{\omega' \in T} \tilde{P}(\xi_O \xi^{\omega'}_{\bar{O}})} \tag{5.3}$$

Algorithm 1 BUILDTABLE (s_1, AA)

for $\omega \in I$ **do**
  $s^\omega, x^\omega = AA(s_1, \xi^\omega)$
return $\{\xi^\omega, s^\omega, x^\omega\}_{\omega \in T}$

This estimator defines a discrete distribution over the offline information I. The chosen marginalization technique guarantees an estimate that is approximately proportional (not approximately equal) to the true $P(\xi_O)$. Hence, we have that:

$$\forall \omega \in I, \quad P(\xi_{\bar{O}}^\omega \mid \xi_O) \propto \tilde{P}(\xi_{\bar{O}}^\omega \mid \xi_O) \tag{5.4}$$

Sampling from I according to 5.3 yields scenarios with a distribution that takes into account the observed values.

3.2 Building a Contingency Table

If a significant amount of time is available in the offline phase, we can exploit the offline information more aggressively, by trying to prepare for each likely future development. Intuitively, we can *treat each scenario $\omega \in I$ as if it were an actual sequence of online observations, and process it via some anticipatory algorithm*. By doing this, we build a pool of solutions that can then be used to guide an online method.

The process is outlined in Algorithm 1, which requires as input the initial state s_1 of the system, and a solution algorithm AA, accepting the same parameters as ANTICIPATE. The result is an augmented version of the offline information, where each scenario ω is additionally associated to the sequence of states s^ω visited by the algorithm and its sequence of decisions x^ω. We refer to this data structure as *contingency table*, and to its elements as *traces*. We denote the table as T.

3.3 Efficient Online Fixing Heuristic

We use the traces from T to guide an efficient *fixing heuristic*, which tries to choose decisions having the largest chance of being optimal. Formally, it solves:

$$\arg\max\{P^*(x_k \mid s_k\xi_O) : x_k \in X_k\} \tag{5.5}$$

where P^* is the probability that the chosen x_k is optimal, given the state s_k and the observed uncertainty. The X_k set represents the feasible decision space, which is defined via problem-dependent constraints and auxiliary variables.

Closed-forms for P^* can be obtained separately for discrete and numeric problems, based on the contingency table. The process is described in detail in Section 4, and relies on several approximations. Overall, *in case of discrete decisions, the problem from 5.5 translates into*:

$$\arg\min\left\{-\sum_{j=1}^{m}\sum_{v\in D_j}\log p_{jv}[\![x_{kj}=v]\!] : x_k \in X_k\right\} \tag{5.6}$$

where $[\![\cdot]\!]$ denotes the truth value of a predicate, D_j is the domain of x_{kj}, and:

$$p_{jv} = \frac{\sum_{\omega\in T, x^{\omega}_{kj}=v} P(\omega)}{\sum_{\omega\in T} P(\omega)} \tag{5.7}$$

Here, $P(\omega)$ is a compact notation for the probability that *we reach the same state as trace ω, and then everything goes according to plan*. It can be approximated using:

$$\forall\omega \in T, \quad P(\omega) \propto \tilde{P}(s^{\omega}_{sk+1} \mid s_k)\tilde{P}(\xi^{\omega}_{\bar{O}} \mid \xi_O) \tag{5.8}$$

where $\tilde{P}(\xi_{\bar{O}} \mid \xi_O)$ is the estimator from 5.3, and $\tilde{P}(s_{sk+1} \mid s_k)$ is a second estimator obtained via similar means. The cost function in 5.6 is linear if a one-hot encoding is adopted for x_{kj}, and the size of T

Algorithm 2 FIXING (s_1, ξ, T)

for $k = 1 \dots n$ **do**
  $O = O \cup peek(s_k)$
  $\Omega =$ top elements in T by descending 5.8
  Compute p_{jv} and/or p_ω based on Ω
  Solve 5.6/(5.9) to obtain x_k
  $s_{k+1} = next(s_k, x_k, \xi_O)$
return s, x

affects only the computation of the p_{jv} values. Overall, the problem is efficient to solve. *In case of numeric decisions, we have instead*:

$$\arg\min \left\{ \sum_{j=1}^{m} \sum_{\omega \in T} p_\omega \frac{1}{2\sigma_j} (x_{kj} - x_{kj}^{\omega})^2 : x_k \in X_k \right\} \tag{5.9}$$

with:

$$p_\omega = \frac{P(\omega)}{\sum_{\omega' \in T} P(\omega')} \tag{5.10}$$

The cost function is quadratic and convex, and the problem size is small due to the same arguments as 5.6.
Intuitively, the discrete version of the heuristic is related to minimizing weighted discrepancies w.r.t. the traces in T, i.e. to weighted Hamming distances. The numeric version is instead related to weighted Euclidean distances. The pseudo-code for the heuristic is provided in Algorithm 2. The only difference with the process described so far is that the p_{jv} and p_ω probabilities may be computed based on a subset Ω of the full contingency table. This may be useful to bias the choice of the online decision according to the most relevant traces.

4 Deriving the FIXING Heuristic

Our main goal will be to obtain a closed-form for P^* in 5.5, which will require several approximations. We start by treating all components in

x_k as statistically independent. This allows to state P^* as a product of $P^*(x_{kj} \mid s_k\xi_O)$ probabilities, related to individual components of x_k. Applying a log transformation then leads to the equivalent problem:

$$\arg\min \left\{ -\sum_{j=1}^{m} \log P^*(x_{kj} \mid s_k\xi_O) : x_k \in X_k \right\} \tag{5.11}$$

where m is the cardinality of x_k. We then assume that a decision x_{kj} is optimal if the current optimization process is similar to a trace in the contingency table, and x_{kj} is similar to the decision made in that circumstance. Formally, we can obtain $P^*(x_{kj})$ via marginalization:

$$P^*(x_{kj} \mid s_k\xi_O) = \frac{\sum_{\omega\in T} P(\omega)P^*(x_{kj} \mid \omega)}{\sum_{\omega\in T} P(\omega)} \tag{5.12}$$

where $P(\omega)$ is compact notation for $P(s^{\omega}_{k+1}\xi^{\omega}_{\bar{O}} \mid \xi_O s_k)$. By assuming independence between the s and ξ variables, and applying the techniques used for 5.3, we get:

$$P(\omega) \propto \tilde{P}(\xi^{\omega}_{\bar{O}} \mid \xi_O)\frac{\tilde{P}(s_k s^{\omega}_{k+1})}{\sum_{\omega'\in T} \tilde{P}(s_k s^{\omega'}_{k+1})} \tag{5.13}$$

where the estimator for $\tilde{P}(s_k s_{k+1})$ can be trained over data from the contingency table. We now need a way to estimate $P^*(x_{kj} \mid \omega)$. *In the discrete case*, we assume that x_{kj} is optimal iff it matches the value from the contingency table, i.e. $P^*(x_{kj} \mid \omega)$ is equal to the truth value of the predicate $x_{ki} = x^{\omega}_{kj}$. Hence, 5.12 becomes:

$$P^*(x_{kj} \mid s_k\xi_O) = \sum_{v\in D_j} p_{jv}[\![x_{kj} = v]\!] \tag{5.14}$$

with p_{jv} as in 5.7. By applying the log transformation, and using the fact that values in D_j are mutually exclusive, we get the discrete formulation from 5.6.

In the numeric case, we assume that decisions close to the one in the trace have a chance of being optimal, which follows a Normal distribution. Formally, we have that:

$$P^*(x_{kj} \mid \omega) = \frac{1}{\sqrt{2\pi}\sigma_j} e^{-\frac{1}{2\sigma_j}(x_{kj} - x^{\omega}_{kj})^2} \tag{5.15}$$

where σ_j is the standard deviation of the value of x_{kj} in the contingency table. By applying the log transformation to 5.12, then Jensen's inequality, and by getting rid of offset terms (which have no impact on optimization), we get the numeric formulation from 5.9. Note that, due to the use of Jensen's inequality, the resulting cost function is actually an approximated upper bound for the original probability.

5 Formal Method Description

Our three solution methods can now be defined with relative ease, by combining the techniques just described.

5.1 ANTICIPATE-D

Our first hybrid method is obtained from ANTICIPATE by simply replacing the static set of samples with a dynamically adjusted one. The dynamic set can be populated according to the estimated probabilities from 5.3, so as not to loose relevance: this may enable to reach similar solution qualities with fewer scenarios, at the cost of training an estimator offline. We refer to this approach as ANTICIPATE-D, and its pseudo-code is in 3

Algorithm 3 ANTICIPATE-D (s_1, ξ)

Train the $\tilde{P}(\xi)$ estimator on I
for $k = 1 \dots n$ **do**
  $O = O \cup peek(s_k)$
  Sample Ω from T, according to 5.3
  $x_k = A(s_k, \xi_O, \{\xi^\omega\}_{\omega \in \Omega})$
  $s_{k+1} = next(s_k, x_k, \xi_O)$
return s, x

5.2 CONTINGENCY

The second method is based on the idea of computing robust solutions for the scenarios in the offline information, and then use them as guidance for the FIXING heuristic. Robust solutions are obtained by using ANTICIPATE, so that hopefully the (fast) fixing heuristic will be able to match their quality: the price to pay is a hefty offline computational effort. We refer to this approach as CONTINGENCY, and its pseudo-code is reported in 4.

Algorithm 4 CONTINGENCY (s_1, ξ)

Train the $\tilde{P}(\xi)$ estimator on I
$T = \text{BUILDTABLE}(s_1, \text{ANTICIPATE})$
Train the $\tilde{P}(s_k s_{k+1})$ estimators on T, for all k
$s, x = \text{FIXING}(s_1, \xi, T)$
return s, x

5.3 CONTINGENCY-D

Algorithm 5 CONTINGENCY-D (s_1, ξ)

Train the $\tilde{P}(\xi)$ estimator on I
$T = \text{BUILDTABLE}(s_1, \text{ANTICIPATE}_1)$
Train the $\tilde{P}(s_k s_{k+1})$ estimators on T, for all k
$s, x = \text{FIXING}(s_1, \xi, T)$
return s, x

Our final method is similar to the previous one, except that the contingency table is populated with *non-robust* solutions. This is done by using ANTICIPATE *with a single scenario, given by the values of* ξ^ω *(i.e. the pretend online observations)*. This technique (referred to as ANTICIPATE_1) provides perfect information about the future, so that achieving robustness is entirely delegated to the FIXING heuristic. The

approach is likely to loose reliability, but has two important advantages: 1) lower offline computational costs; and 2) while ANTICIPATE is a stochastic algorithm, ANTICIPATE_1 is deterministic. So, this method may provide anticipatory-like results even when no anticipatory algorithm is available. We refer to this method as CONTINGENCY-D, and its pseudo-code is reported in Algorithm 5.

Chapter 6

Instantiating the Methods

Grounding our approaches requires to specify: 1) the x, s and ξ variables, 2) the $peek$ and $next$ functions, 3) the sampling-based algorithm A, and 4) the feasible space X_k for the FIXING heuristic. Additionally, evaluating the solution quality requires to define 5) a cost metric.

We show how this can be done in two case studies: 1) a Virtual Power Plant energy management problem with numerical decisions; and 2) a combinatorial Traveling Salesman Problem with uncertain travel times. In both cases, the input anticipatory algorithm A is given by a Mathematical Programming model, based on the Sample Average Approximation. The models are slight improvements over those by [DFLM18b], whose work brought to attention the interplay between offline and online phases. Both approaches are serviceable, but not necessarily representative of the state-of-the-art (especially for the TSP). It would be useful also to underline that a MPC-like approach would be similar to ANTICIPATE and ANTICIPATE-D, but where a single scenario is used, which corresponds to the expected value of the random variables/distributions. In particular, in the VPP case study we assume that the prediction error follow roughly a Normal distribution $N(0, \sigma^2)$ so a MPC-like approach is exactly our ANTICIPATE and ANTICIPATE-D with a single scenario. Such an algorithm nicely spans the space between a myopic heuristic, and a two-stage anticipatory algorithm with

many scenarios: in terms of both online computation and the quality of outcome. For the TSP case study it is different because we need to solve our model with a pool of scenarios to obtain the expected values that compose our single scenario for the MPC approach.

1 Instantiating the Methods for the VPP Energy Problem

A Virtual Power Plant aggregates different sources of power generation and consumption to offer a predictable power envelope. Managing a VPP requires to route power flows so as to satisfy the demand, to obey physical limits, and to minimize the operating costs [PBBA$^+$11b, BMRY15]. Both the demand and the RES generation are uncertain.

1.1 Instantiating the Baseline Model

Formally, the decision vector x_k specifies the power flow x_{kj} to/from each node (demand, generator, storage…). In particular, we assume that x_{kS} refers to flow for the storage system. The state component s_{kS} corresponds to the storage charge level, while s_{kD} to its flow direction. The random variable ξ_{kL} corresponds to the load, while ξ_{kR} to the RES generation. The $peek$ function simply returns the pairs (k, L) and (k, R). The $next$ function is given by:

$$s_{k+1,S} = s_k + \eta x_{kS} \tag{6.1}$$

$$s_{k+1,D} = 0 \text{ if } x_{kS} \leq 0 \text{ and 1 otherwise} \tag{6.2}$$

where η is the charging efficiency of the storage system. The feasible space X_k is given by the Mathematical Program:

$$\xi_{kL} = \sum_{j=1}^{m} x_{kj} + \xi_{kR} \tag{6.3}$$

$$l_j \leq x_{kj} \leq u_j \qquad \forall j \in [1..m] \tag{6.4}$$

$$0 \leq s_k + \eta x_{kS} \leq \Gamma \tag{6.5}$$

$$x_k \in \mathbb{R}^m \tag{6.6}$$

where Eq. (6.3) enforces power balance, Eq. (6.4) states the physical limits for the power flows, and Eq. (6.5) those for the storage charge. The cost incurred at each stage is given by:

$$\sum_{j=1}^{m} c_{kj} x_{kj} + \alpha |s_{kD} - s_{k+1,D}| \tag{6.7}$$

where c_{kj} is a cost associate to each flow. Unlike the model from [DFLM18b], we include a cost term α related to storage wear-off, which increases each time the corresponding flow switches direction. Due to this term, the input algorithm A needs to solve an NP-hard problem, while the fixing heuristic has no such need.

1.2 The Models of Uncertainty

The models of uncertainty for both cases studies are technically mixtures of Gaussians. They are designed first to ensure a realistic level of dependence between the random variables, and second for simplicity. For the VPP, we assume that both the RES power generation and the load at each stage may exhibit Normally distributed deviations from a number of different possible behaviors. Formally, each mean and standard deviation is controlled by a second "mode" random variable ψ. Using the RES generation ξ_{kR} as an example, we have that:

$$\xi_{kR} \sim \mathcal{N}(\mu_{k\psi}, \sigma_{k\psi}) \qquad \forall k \in [1..n] \tag{6.8}$$

The ψ variable is integer-valued and acts as an index to specify which mean and standard deviation should be used at each stage. In other words, ψ controls which component of a Gaussian mixture is used to generate the data. We assume that ψ follows a discrete uniform distribution.

Since all stages rely on the same ψ variable, the mixtures are synchronized: this ensures statistical dependence and simplifies the definition of reasonable parameters. In particular, we chose our μ vectors so that the μ_{ki} values related to the same i index correspond respectively to historical daily records of energy production [SE07] and aggregated load [GYI02]. Each σ_{ki} value is set to $0.2\mu_{ki}$.

1.3 Instantiating ANTICIPATE

In detail, the base algorithm A for the VPP requires to solve a mathematical program, which is best described by grouping its equations in blocks.

The main decision variables are the power flows to/from the m nodes in the system (except the RES generators and the demand/load), with a positive flow meaning that energy is routed towards the node. There is one x_{kj} variable for each node j related to the current stage k, and (as in all SAA approaches) one copy x^{ω}_{hj} per scenario ω of the flow variables related to future stages (i.e. with $h > k$).

Auxiliary variables y_k (for the current stage) and y^{ω}_{k} (for the scenarios) are used to keep track of flow inversions for the storage system, which are linked to wear-off effects. With these variables, we can define the problem objective, which is the (approximate) expected cost at the current stage:

$$\min f_k(x, y) + \frac{1}{|\Omega|} \sum_{\omega \in \Omega} f_\omega(x, y) \tag{6.9}$$

with the current stage cost being given by:

$$f_k(x, y) = \sum_{j=1}^{m} c_{kj} x_{kj} + \alpha y_k \tag{6.10}$$

and the cost for future stages in scenario ω by:

$$f_\omega(x, y) = \sum_{h=k+1}^{eoh} \sum_{j=1}^{m} c_{hj} x_{hj}^{\omega} + \alpha y_h^{\omega} \tag{6.11}$$

where c_{kj} is the cost associated to flow j at stage k (e.g. the cost of buying energy from the grid), and α is the cost of one flow inversion. The eoh parameter defines how many future stage are taken into account by the anticipatory algorithm, and it is the minimum between the look-ahead horizon and the number of stages n.
The flows at each stage and scenario should obey their respective physical limits. Hence we have, for $j \in [1..m]$:

$$l_j \leq x_{kj} \leq u_j \tag{6.12}$$

$$l_j \leq x_{hj}^{\omega} \leq u_j \qquad \forall \omega \in \Omega, \forall h \in [k+1..eoh] \tag{6.13}$$

Power balance should be reached for each stage and scenario. This condition is dependent on the uncertain value of the load ξ_{kL} (ξ_{hL}^{ω} for the scenarios), and the power from Renewable Energy Sources ξ_{hR} (ξ_{kR}^{ω} for the scenarios):

$$\xi_{kL} = \sum_{j=1}^{m} x_{kj} + \xi_{kR} \tag{6.14}$$

$$\xi_{hL}^{\omega} = \sum_{j=1}^{m} x_{hj}^{\omega} + \xi_{kR}^{\omega} \;\; \forall \omega \in \Omega, \forall h \in [k+1..eoh] \tag{6.15}$$

The flow to/from the storage system affects its charge level s_{kS} (s_{hS}^{ω} for the scenarios). Hence we have for $\omega \in \Omega$:

$$s_{k+1,S}^{\omega} = s_{k,S} + \eta x_{kS} \tag{6.16}$$

$$s_{h+1,S}^{\omega} = s_{h,S}^{\omega} + \eta x_{hS}^{\omega} \qquad \forall h = [k+1..eoh] \tag{6.17}$$

The charge level of the storage system s^{ω}_{hS} in the scenarios should be within its physical limits for each stage (for stage k, the level is assumed to be fine):

$$0 \leq s^{\omega}_{hS} \leq \Gamma \qquad \forall \omega \in \Omega, \forall h \in [k+1..eoh+1] \tag{6.18}$$

A last set of constraints is necessary to link the inversion variables y_k (y^{ω}_k for the scenarios) with the flow variables. This is done in two steps[1]: first, we use auxiliary binary variable s^{ω}_{hD} to track the direction of the flow to/from the storage system. In particular, we have that $s^{\omega}_{hD} = 0$ if power was drained from the storage in the *previous* stage. For stage k, we have an s_{kD} parameter with the same semantic. Overall, we have:

$$y_k \geq s^{\omega}_{k+1,D} - s_{kD} \qquad \forall \omega \in \Omega \tag{6.19}$$

$$y_k \geq s_{kD} - s^{\omega}_{k+1,D} \qquad \forall \omega \in \Omega \tag{6.20}$$

for stage k. For the subsequent stages and scenarios:

$$y^{\omega}_h \geq s^{\omega}_{h+1,D} - s^{\omega}_{hD} \qquad \forall \omega \in \Omega, \forall h \in [k+1..eoh] \tag{6.21}$$

$$y^{\omega}_h \geq s^{\omega}_{hD} - s^{\omega}_{h+1,D} \qquad \forall \omega \in \Omega, \forall h \in [k+1..eoh] \tag{6.22}$$

Second, we link the s^{ω}_{hD} variables to the flows by means of indicator constraints. For each scenario $\omega \in \Omega$ we have:

$$s^{\omega}_{k+1,D} \Rightarrow x_{kS} \leq 0 \tag{6.23}$$

$$1 - s^{\omega}_{k+1,D} \Rightarrow x_{kS} \geq 0 \tag{6.24}$$

$$s^{\omega}_{h+1,D} \Rightarrow x^{\omega}_{hS} \leq 0 \qquad \forall h \in [k+1..eoh] \tag{6.25}$$

$$1 - s^{\omega}_{h+1,D} \Rightarrow x^{\omega}_{hS} \geq 0 \qquad \forall h \in [k+1..eoh] \tag{6.26}$$

Solving (6.9)-(6.26) yields set of values for the flow variables related to stage k, which approximately optimize the expected cost of operations.

[1]This part of the model has been altered w.r.t. the implementation, for sake of keeping the notation consistent with Section 4

1.4 Instantiating ANTICIPATE-D

Our first hybrid method is obtained from ANTICIPATE by simply replacing the static set of samples with a dynamically adjusted one.
The dynamic set can be populated according to the estimated probabilities from 5.3, so as not to loose relevance: this may enable to reach similar solution qualities with fewer scenarios, at the cost of training an estimator offline.
We use Kernel Density Estimation (KDE with Gaussian Kernels) to obtain all approximate distributions.

1.5 Instantiating CONTINGENCY

The second method is based on the idea of computing robust solutions for the scenarios in the offline information, and then use them as guidance for the FIXING heuristic.
Robust solutions are obtained by using ANTICIPATE, so that hopefully the (fast) fixing heuristic will be able to match their quality: the price to pay is a hefty offline computational effort. We refer to this approach as CONTINGENCY.

1.6 Instantiating CONTINGENCY-D

Our final method is similar to the previous one, except that the contingency table is populated with *non-robust* solutions. This is done by using ANTICIPATE *with a single scenario, given by the values of* ξ^{ω} *(i.e. the pretend online observations)*. This technique (referred to as ANTICIPATE$_1$) provides perfect information about the future, so that achieving robustness is entirely delegated to the FIXING heuristic.
The approach is likely to loose reliability, but has two important advantages:

1. lower offline computational costs

2. while ANTICIPATE is a stochastic algorithm, ANTICIPATE_1 is deterministic.

So, this method may provide anticipatory-like results even when no anticipatory algorithm is available. We refer to this method as CONTINGENCY-D.

2 Results for the VPP

We empirically evaluated the three hybrid offline/online methods on realistic instances for the case study. The baseline is a myopic heuristic.

2.1 Experimental Setup

Our methods are evaluated over different uncertainty realizations, obtained by sampling the random variables for the loads and RES generation in the VPP. We use models of uncertainty that ensure realistic statistical dependence between the variables (see 1.2). This process yields the offline information I and the sequences of observations for the experiments.

For the VPP, grid electricity prices change every 15 minutes, which is also the duration of our online stages. New offline information (e.g. market prices) becomes available every day, hence our horizon corresponds to $24 \times 4 = 96$ stages. We use (real) physical bounds for power generation from [BMRY15, EO15]. The initial battery state, efficiency, and power flow limit, etc. are also based on real data [BMRY15, EO15]. Different instances have then been obtained by manually scaling load and RES generation.

We use Kernel Density Estimation (with Gaussian Kernels) to obtain all approximate distributions. As an underlying solver we use Gurobi [2], which can handle both MILPs and Quadratic Programs. Each evaluated algorithm and configuration is run 50 times, with the same 50

[2] Available at `http://www.gurobi.com`

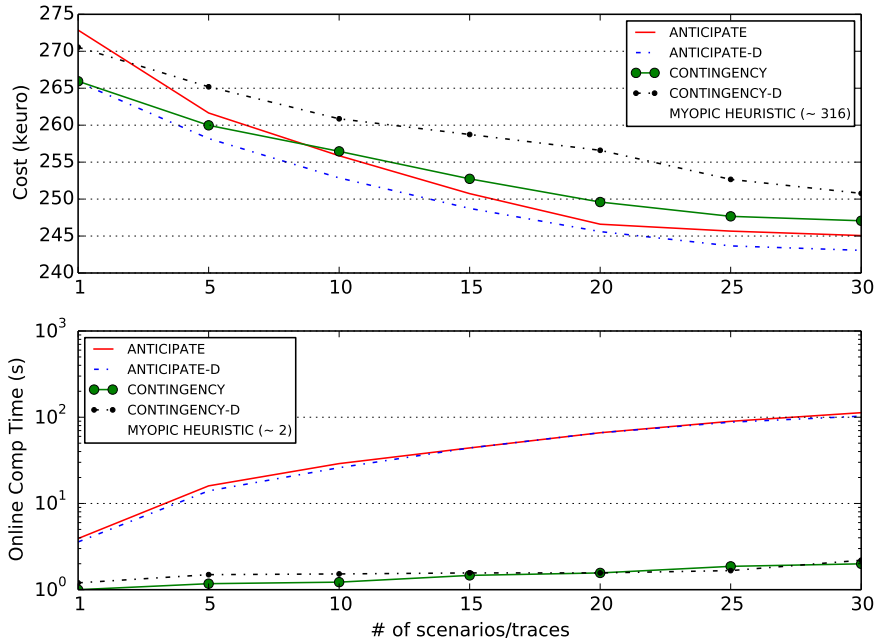

Figure 6.1: Methods solution/quality comparison for the VPP

sequences of realizations. We use a time limit of 300 seconds. For each run we record both the time required by each approach and the corresponding solution cost, and we report their average values over the 50 realizations. In all cases, $|I| = |T| = 100$, and for the CONTINGENCY method, the contingency table is built using ANTICIPATE with 20 scenarios.

2.2 Discussion

The offline training times of the KDE models are roughly the same for all the three hybrid methods ($\sim$ 65 sec for the VPP). Building the contingency tables for CONTINGENCY takes $\sim 6,000$ sec in the VPP, but only ~ 400 sec for CONTINGENCY-D.

In Figure 6.1 we show the cost/quality tradeoff of the proposed methods and of ANTICIPATE for the VPP (base instance). The use of a dynamic set of scenarios allows ANTICIPATE-D to work better than ANTICIPATE.

Method	VPP (σ)
Myopic H	8.499
ANTICIPATE	4.994
ANTICIPATE-D	5.730
CONTINGENCY	5.557
CONTINGENCY-D	7.017

Table 6.1: Standard deviation comparison for the VPP

The CONTINGENCY method is surprisingly close in terms of quality to the original anticipatory algorithm, especially considered its dramatically smaller online computational cost (up to two orders of magnitude). CONTINGENCY-D performs slightly worse than CONTINGENCY, but it still much better than the myopic heuristic. Increasing the number of guiding traces is beneficial in particular for CONTINGENCY-D. We also show that a MPC-like approach is exactly our ANTICIPATE and ANTICIPATE-D with a single scenario. Such an algorithm nicely spans the space between a myopic heuristic, and a two-stage anticipatory algorithm with many scenarios: in terms of both online computation and the quality of outcome.
In Table 6.1 we show the standard deviation for the solution quality over the 50 realizations with 20 scenarios/traces (on the same instances as Figure 6.1). All values are significantly lower than the quality gap with the myopic heuristic. As expected CONTINGENCY-D tends to be less stable than the other methods due to its reliance on non-robust guiding traces.

3 The Traveling Salesman Problem Case Study

As a second case study, we consider a TSP over an asymmetric, fully connected, graph with uncertain, exogenous, travel times (e.g. the visits have a negligible impact on traffic).

3.1 Instantiating the Baseline Model

In this case, the x_k vector includes m components x_{kj}, each equal to 1 iff j is the next node to be visited. There is no x_{kj} variable for the depot, which is reached by default once all other nodes are visited. The state vector contains a component s_{kj} equal to 1 iff node j (excluding the depot) has been visited at stage k, plus a s_{kC} component specifying the index of the current node. The uncertainty is modeled via random variables ξ_{ij}, each associated to the travel time between nodes i and j. The travel times for all outgoing arcs from i are observed when the node is visited, i.e. the $peek$ function returns the pairs (s_{kC}, j) with $j \in [1..m]$. The $next$ function is:

$$s_{k+1,C} = \sum_{j=1}^{m} j\, x_{kj} \tag{6.27}$$

$$s_{k+1,j} = \max(x_{kj}, s_{k,j}) \qquad \forall j \in [1..m] \tag{6.28}$$

where 6.27 makes sure that the value $s_{k+1,C}$ matches the index of the next node to be visited. The feasible space X_k is given by the Mathematical Program:

$$\sum_{j=1}^{m} x_{kj} = 1 \tag{6.29}$$

$$x_{kj} \leq 1 - s_{kj} \qquad \forall j \in [1..m] \tag{6.30}$$

which forces moving to a single, unvisited node. The cost incurred at each stage is the travel time to the next node, i.e.:

$$\sum_{j=1}^{m} \xi_{s_{kC},j}\, x_{kj} \tag{6.31}$$

The final cost is obtained by summing the cost of each stage, plus the distance from the last visited node to the depot. Once again, while the anticipatory algorithm A needs to solve an NP-hard problem (stochastic TSP), the fixing heuristic has no such need and is therefore much faster.

3.2 The Models of Uncertainty

Our model of uncertainty for the TSP is based on similar ideas, but makes use of a more complex sampling process. First, we assume that the travel times of all arcs from a given node i follow a Normal distribution, whose mean and variance is controlled by an additional (binary) random variable ψ_i. Formally, we have that:

$$\xi_{ij} \sim \mathcal{N}((1 + \psi_i)\mu_{ij}, 0.1\mu_{ij}) \tag{6.32}$$

If we sample 1 from ψ_i, the travel times of all outgoing arcs become twice as large on average. The μ_{ij} values correspond to the deterministic distances from classical asymmetrical TSP benchmarks.

This approach ensures statistical dependence between multiple arcs from the same node, but not between the nodes themselves. This is unrealistic (nearby nodes tend to become congested at the same time), and an issue for our experimentation: in fact, unless observing the travel times for some node i provides information for some other node j, the ANTICIPATE-D method can provide no benefit.

We obtain realistic dependence between nodes by sampling the ψ_i variables according to a stochastic, time-discrete, dynamic process. Namely, we assume that the distribution of the ψ_i variables evolves over a sequence of discrete steps. In particular, let ψ_i^k be the ψ_i variable at the k-th step of its evolution. The chance that $\psi_i^k = 1$ (i.e. that node i is congested) is given by:

$$P(\psi_i^k = 1) = \beta + (1 - 2\beta)p_i^k \tag{6.33}$$

with

$$p_i^k = \alpha\psi_i^{k-1} + (1 - \alpha)\frac{1}{Z_i}\sum_{j \neq i} e^{-\mu_{ij}}\psi_j^{k-1} \tag{6.34}$$

$$Z_i = \sum_{j \neq i} e^{-\mu_{ij}} \tag{6.35}$$

where $\alpha, \beta \in [0, 1]$. 6.34 ensures that: 1) a node that is congested at step $k-1$ has increased likelihood (measured by α) of being congested

at step k; and 2) the presence of nearby congested nodes at step $k-1$ increases the chance that i is congested at step k – the intensity of this correlation decreases with the distance, following an exponential law. 6.33 ensures that the probability that node i is congested is never smaller than β, and never larger than $1-\beta$.

We can simulate the process by sampling ψ_i^0 according to independent Bernoulli distributions, and then using 6.33 to sample for any number of steps. After a few iterations, the values of the ψ_i^k variables will exhibit some degree of correlation, which can be tuned by choosing the values of α and β. We sample our ψ_i variables by drawing ψ^k vectors from this process uniformly at random.

Figure 6.2 shows the cross-correlation matrix for 1,000 iterations of the process[3], $\alpha = 0.5$, $\beta = 0.1$, on a six-node TSP instance. Darker cells denote a higher cross-correlation, and the self-correlation is 1 by construction (which explains the black diagonal). The correlations are stable and exhibit only moderate variations over multiple runs of the process.

3.3 Instantiating ANTICIPATE

The base algorithm A for the TSP is also based on a mathematical program. At stage k, the model has a binary decision variable x_{kj} for each of the $m+1$ nodes in the graph (node 0 is the depot), plus variables x_{ij}^{ω} for each pair of nodes and for each scenario. We have that $x_{kj} = 1$ iff, after the current node s_{kC}, we move to node j. We have that $x_{ij}^{\omega} = 1$ iff the arc from node i to node j is part of the route for scenario ω.

The cost function is the (approximate) expected travel distance, which is given by:

$$\min \sum_{j=0}^{m} \xi_{s_{kC},j} x_{kj} + \frac{1}{|\Omega|} \sum_{\omega \in \Omega} \sum_{i=0}^{m} \sum_{j=0}^{m} \xi_{ij} x_{ij}^{\omega} \tag{6.36}$$

[3] Actually 1,100 iterations, but the first 100 are discarded to avoid tail-in effects.

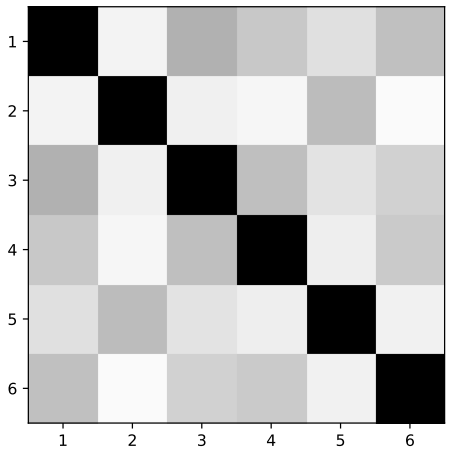

Figure 6.2: Cross-correlation matrix for the ψ_i variables in a 6-node TSP instance

where $\xi_{s_{kC},j}$ is the observed travel time from the current node s_{kC} to node j, and ξ_{ij}^{ω} is the travel time between nodes i and j in scenario ω. Nodes already visited cannot be visited again. This is enforced by relying on a set of parameters s_{kj}, such that $s_{kj} = 1$ iff node j has been visited at stage k. The current node is always visited, while the depot never counts as visited (and hence does not need a corresponding parameter). Overall:

$$x_{kj} \leq 1 - s_{kj} \qquad \forall j \in [1..m] \tag{6.37}$$

$$x_{ij}^{\omega} \leq 1 - s_{kj} \qquad \forall \omega \in \Omega, \forall i, j \in [1..m] \tag{6.38}$$

Each non-visited node must be the successor of some other node in each scenario:

$$\sum_{i=0}^{m} x_{ij}^{\omega} = 1 \qquad \forall \omega \in \Omega, \forall j \in [0..m] \tag{6.39}$$

There can be single successor for the for the current stage:

$$\sum_{j=0}^{m} x_{kj} = 1 \tag{6.40}$$

Each node *except the depot* must have exactly one successor in each scenario:

$$\sum_{j=0}^{m} x_{ij}^{\omega} = 1 \qquad \forall \omega \in \Omega, \forall i \in [1..m] \tag{6.41}$$

Sub-loops are prevented by associating a sequence variable $t_i^{\omega} \geq 0$ to each node in each scenario[MTZ60], and then enforcing, for each scenario $\omega \in \Omega$:

$$t_{s_{kC}}^{\omega} = 0 \tag{6.42}$$

$$t_j^{\omega} \geq t_i^{\omega} - m + (m+1)x_{ij}^{\omega} \qquad \forall i, j \in [0..m] \tag{6.43}$$

Solving 6.36-(6.43) yields a successor for the current node that approximately optimizes the expected travel time.
To instantiate ANTICIPATE-D, CONTINGENCY and CONTINGENCY-D for the TSP problem, we can refer to Subsections from 1.4 to 1.6 based on the previous instantiations of the Baseline Model and the ANTICIPATE method.

4 Results for the TSP

We empirically evaluated the three hybrid offline/online methods on realistic instances. The baseline is a myopic heuristic.

4.1 Experimental Setup

Our methods are evaluated over different uncertainty realizations, obtained by sampling the random variables the travel times in the TSP. We use models of uncertainty that ensure realistic statistical dependence between the variables. This process yields the offline information I and the sequences of observations for the experiments.
For the TSP we use classical benchmarks[4], by including problems from

[4] http://myweb.uiowa.edu/bthoa/TSPTWBenchmarkDataSets.htm

10 to 40 nodes. In the TSP each stage represents a visit, hence our horizon corresponds to the total number of nodes.

We use Kernel Density Estimation (with Gaussian Kernels) to obtain all approximate distributions. As an underlying solver we use Gurobi [5], which can handle both MILPs and Quadratic Programs. Each evaluated algorithm and configuration is run 50 times, with the same 50 sequences of realizations. We use a time limit of 300 seconds. For each run we record both the time required by each approach and the corresponding solution cost, and we report their average values over the 50 realizations. In all cases, $|I| = |T| = 100$, and for the CONTINGENCY method, the contingency table is built using ANTICIPATE with 20 scenarios.

4.2 Discussion

The offline training times of the KDE models are roughly the same for all the three hybrid methods ($\sim$ 32 sec for the TSP). Building the contingency tables for CONTINGENCY takes $\sim 15,000$ sec in the TSP, but only $\sim 2,000$ sec for CONTINGENCY-D.

In Figure 6.3 we show the cost/quality tradeoff of the proposed methods and of ANTICIPATE for the TSP (a representative 20 customers instance). The use of a dynamic set of scenarios allows ANTICIPATE-D to work better than ANTICIPATE. The CONTINGENCY method is surprisingly close in terms of quality to the original anticipatory algorithm, especially considered its dramatically smaller online computational cost (up to two orders of magnitude). CONTINGENCY-D performs slightly worse than CONTINGENCY, but it still much better than the myopic heuristic. Increasing the number of guiding traces is beneficial in particular for CONTINGENCY-D. It would be useful, also in this case study, to underline that a MPC-like approach would be similar to ANTICIPATE and ANTICIPATE-D, but where a single scenario is used, which corre-

[5] Available at `http://www.gurobi.com`

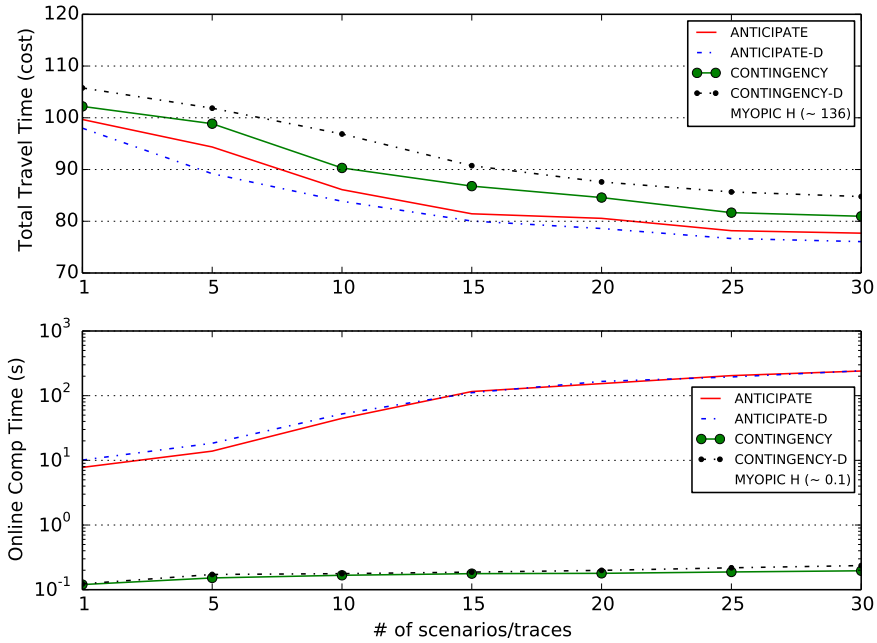

Figure 6.3: Methods solution/quality comparison for the TSP

sponds to the expected value of the random variables/distributions. In particular, for the TSP case study we need to solve (offline) our model with a pool of scenarios to obtain the expected values that compose our single scenario for the MPC approach.

Method	**TSP** (σ)
Myopic H	7.106
ANTICIPATE	1.889
ANTICIPATE-D	2.846
CONTINGENCY	3.788
CONTINGENCY-D	5.934

Table 6.2: Standard deviation comparison for TSP

In Table 6.2 we show the standard deviation for the solution quality over the 50 realizations with 20 scenarios/traces for the TSP (on the

same instances as Figure 6.3). All values are significantly lower than the quality gap with the myopic heuristic. As expected CONTINGENCY-D tends to be less stable than the other methods due to its reliance on non-robust guiding traces.

Chapter 7

Concluding Remarks & Future Works

This thesis makes a first significant step toward generic integrated offline/online optimization under uncertainty. We propose two groups of methods that represent two distinct (but very related) contributions.

On one hand, the thesis focuses on the idea that many practical application scenarios require to make *interdependent offline and online decisions*. For example, we may need to define a daily production schedule for an industrial plant, and then manage its power supply on a hour by hour basis; or we may assign customers to vehicles for delivering goods, and then adjust their routes dynamically as the traffic conditions reveal themselves over time. The simplest approach to tackle such problems is to deal with the offline and online phase separately, respectively (e.g.) via a sampling-based method and a heuristic. However, we showed that *substantial improvements can be obtained by treating the two phases in an integrated fashion*.

On the other hand, we start from the idea that online anticipatory algorithms often have a considerable computational cost, which may be problematic if online decisions must be taken within a short time frame. In most practical settings, however, a substantial amount of time and information is available before the online problem is solved, in an *of-*

fline phase. For example, one may have access to energy production forecasts, historical travel times in routing problems, results from test runs in cyber-physical systems. We refer to this sort of data as *offline information*. Usually, it is employed to characterize the uncertain elements and for sampling likely outcomes (i.e. scenarios). We showed how to exploit this information at a much deeper level. In this context, we propose three hybrid offline/online methods that build over a given, sampling-based, anticipatory algorithm, and allow to match its solution quality at a fraction of the online computational cost. One of them can even rely on a deterministic algorithm, thus providing state-of-the art performance in problems for which no anticipatory approach is available. All our methods work by shifting part of the computation to the offline phase, where time limits are more relaxed and the costs can be better amortized (e.g. via parallelization).

In the first part, we propose four alternative approaches based on the idea of making the offline and online solvers operate synergistically. In the ANTICIPATE method this is done by providing the online solver with the approximation of an oracle (i.e. replacing the greedy heuristic with a sampling-based anticipatory algorithm). However, increasing the computational load of the online phase may not a good idea when stringent time constraints exist. In such a situation, it may be better to improve the greedy heuristic by simply adjusting its parameters. This is the main idea in the ACKNOWLEDGE approach which maintains the efficiency of the original greedy heuristic, at the price of a computationally expensive parameter tuning process, which is however performed offline. In the remaining methods, we instead make the offline solver aware of the limitations of the online one (i.e. TUNING), and capable of controlling its behavior by adjusting parameters (i.e. ACTIVE). Indeed, shifting our attention to the offline decision, we can mitigate the discrepancy by translating the online greedy heuristic as a set of constraints, which can be injected in the offline model. All the proposed techniques yield substantially improved solutions: ANTICIPATE matches the quality level

of EXPECTATION, but it is applicable under more general assumptions. Unfortunately, the method is also less efficient. ACTIVE often manages to beat ANTICIPATE (and therefore EXPECTATION) in terms of solution quality. While this comes at the price of a substantially increased offline computation time, the method achieves these results by using naive and very efficient online heuristics. We believe there is room for improving the efficiency of our methods (similarly to how EXPECTATION was improved in REGRET), and achieving this goal is part of our current research directions. We also plan to apply our approaches to different problems, such as resource allocation and scheduling with Simple Temporal Networks under Uncertainty.

In the second part of the thesis we have presented three methods that can be applied *to a generic anticipatory algorithm* to reduce its online computational effort by exploiting offline information. In particular, both CONTINGENCY and CONTINGENCY-D are dramatically faster than ANTICIPATE during online operation. Between the two of them CONTINGENCY is significantly more reliable in terms of quality, but may require a substantial amount of offline computation. The ANTICIPATE-D technique provides a modest advantage in terms of solution time, but can match and even surpass ANTICIPATE in terms of quality. The ability to shift part of the computational cost to an offline stage provides a significant degree of flexibility to stochastic anticipatory algorithm, and likely to increase their applicability. We believe there is room for improving the scalability and efficiency of our methods, and achieving this goal is part of our current research directions.

We also plan to apply our approaches to different application problems to demonstrate the broad applicability of our integrated offline/online methods.

Bibliography

[AP12] Shimon Awerbuch and Alistair Preston. *The virtual utility: Accounting, technology & competitive aspects of the emerging industry*, volume 26. Springer Science & Business Media, 2012.

[BBC11] Dimitris Bertsimas, David B Brown, and Constantine Caramanis. Theory and applications of robust optimization. *SIAM review*, 53(3):464–501, 2011.

[BD12] Christina Büsing and Fabio D'Andreagiovanni. New results about multi-band uncertainty in robust optimization. In *International Symposium on Experimental Algorithms*, pages 63–74. Springer, 2012.

[Bel58] Richard Bellman. Dynamic programming and stochastic control processes. *Information and control*, 1(3):228–239, 1958.

[BEY05] Allan Borodin and Ran El-Yaniv. *Online computation and competitive analysis*. cambridge university press, 2005.

[BL97] John R. Birge and Francois Louveaux. Introduction to stochastic programming. series in operations research and financial engineering, 1997.

[BM99] Alberto Bemporad and Manfred Morari. Robust model predictive control: A survey. In *Robustness in identification and control*, pages 207–226. Springer, 1999.

[BMRY15] Hao Bai, Shihong Miao, Xiaohong Ran, and Chang Ye. Optimal dispatch strategy of a virtual power plant containing battery switch stations in a unified electricity market. *Energies*, 8(3):2268–2289, 2015.

[BS04] Dimitris Bertsimas and Melvyn Sim. The price of robustness. *Operations research*, 52(1):35–53, 2004.

[BSL96] Dimitris J Bertsimas and David Simchi-Levi. A new generation of vehicle routing research: robust algorithms, addressing uncertainty. *Operations Research*, 44(2):286–304, 1996.

[BTBB10] Aharon Ben-Tal, Dimitris Bertsimas, and David B Brown. A soft robust model for optimization under ambiguity. *Operations research*, 58(4-part-2):1220–1234, 2010.

[BTN08] Aharon Ben-Tal and Arkadi Nemirovski. Selected topics in robust convex optimization. *Mathematical Programming*, 112(1):125–158, 2008.

[BVH04a] Russell Bent and Pascal Van Hentenryck. Online stochastic and robust optimization. In *Annual Asian Computing Science Conference*, pages 286–300. Springer, 2004.

[BVH04b] Russell Bent and Pascal Van Hentenryck. Regrets only! online stochastic optimization under time constraints. In *AAAI*, volume 4, pages 501–506, 2004.

[BVH04c] Russell Bent and Pascal Van Hentenryck. Regrets only! online stochastic optimization under time constraints. In *AAAI*, volume 4, pages 501–506, 2004.

[BVH04d] Russell Bent and Pascal Van Hentenryck. Scenario-based planning for partially dynamic vehicle routing with stochastic customers. *Operations Research*, 52(6):977–987, 2004.

[BZ11] Aqeel Ahmed Bazmi and Gholamreza Zahedi. Sustainable energy systems: Role of optimization modeling techniques

in power generation and supply?a review. *Renewable and Sustainable Energy Reviews*, 15(8):3480 – 3500, 2011.

[CBRJ15] Juan Clavier, Franois Bouffard, Dmitry Rimorov, and Gza Jos. Generation dispatch techniques for remote communities with flexible demand. *IEEE Transactions on Sustainable Energy*, 6(3):720–728, 2015.

[CSS07] Xin Chen, Melvyn Sim, and Peng Sun. A robust optimization perspective on stochastic programming. *Operations Research*, 55(6):1058–1071, 2007.

[DDH+07] Erik Demeulemeester, Filip Deblaere, Jada Herbots, Olivier Lambrechts, and Stijn Van de Vonder. A multi-level approach to project management under uncertainty. *Tijdschrift voor economie en management*, (3):391–409, 2007.

[DFLM17] Allegra De Filippo, Michele Lombardi, and Michela Milano. User-aware electricity price optimization for the competitive market. *Energies*, 10(9):1378, 2017.

[DFLM18a] Allegra De Filippo, Michele Lombardi, and Michela Milano. Methods for off-line/on-line optimization under uncertainty. In *IJCAI*, pages 1270–1276, 2018.

[DFLM18b] Allegra De Filippo, Michele Lombardi, and Michela Milano. Methods for off-line/on-line optimization under uncertainty. In *IJCAI*, 2018.

[DFLM18c] Allegra De Filippo, Michele Lombardi, and Michela Milano. Off-line and on-line optimization under uncertainty: A case study on energy management. In *International Conference on the Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, pages 100–116. Springer, 2018.

[DFLM19] Allegra De Filippo, Michele Lombardi, and Michela Milano. How to tame your anticipatory algorithm. In *Proceedings of*

the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, pages 1071–1077. International Joint Conferences on Artificial Intelligence Organization, 7 2019.

[DFLMB17] Allegra De Filippo, Michele Lombardi, Michela Milano, and Alberto Borghetti. Robust optimization for virtual power plants. In *Conference of the Italian Association for Artificial Intelligence*, pages 17–30. Springer, 2017.

[dMB14] Tito Homem de Mello and Güzin Bayraksan. Monte carlo sampling-based methods for stochastic optimization. 2014.

[DSDB16] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

[ENP12] Richard E. Edwards, Joshua New, and Lynne E. Parker. Predicting future hourly residential electrical consumption: A machine learning case study. *Energy and Buildings*, 49:591 – 603, 2012.

[EO15] Alejandro Navarro Espinosa and L Ochoa. Dissemination document low voltage networks models and low carbon technology profiles. *The University of Manchester*, 2015.

[FM09] Matteo Fischetti and Michele Monaci. Light robustness. In *Robust and online large-scale optimization*, pages 61–84. Springer, 2009.

[FW98] Amos Fiat and Gerhard J Woeginger. *Online algorithms: The state of the art*, volume 1442. Springer, 1998.

[GL94] Jean-Luc Gauvain and Chin-Hui Lee. Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains. *IEEE transactions on speech and audio processing*, 2(2):291–298, 1994.

[GYdH15] Bram L Gorissen, İhsan Yanıkoğlu, and Dick den Hertog. A practical guide to robust optimization. *Omega*, 53:124–137, 2015.

[GYI02] Satoshi Gamou, Ryohei Yokoyama, and Koichi Ito. Optimal unit sizing of cogeneration systems in consideration of uncertain energy demands as continuous random variables. *Energy Conversion and Management*, 43(9):1349 – 1361, 2002.

[HB09] Pascal Van Hentenryck and Russell Bent. *Online stochastic combinatorial optimization*. The MIT Press, 2009.

[HHLBS09] Frank Hutter, Holger H Hoos, Kevin Leyton-Brown, and Thomas Stützle. Paramils: an automatic algorithm configuration framework. *Journal of Artificial Intelligence Research*, 36:267–306, 2009.

[HHLW07] Erwin W Hans, Willy Herroelen, Roel Leus, and Gerhard Wullink. A hierarchical approach to multi-project planning under uncertainty. *Omega*, 35(5):563–577, 2007.

[HL05] Willy Herroelen and Roel Leus. Project scheduling under uncertainty: Survey and research potentials. *European Journal of Operational Research*, 165(2):289 – 306, 2005. Project Management and Scheduling.

[HLM+12] Bri-Mathias Hodge, Debra Lew, Michael Milligan, Hannele Holttinen, Samuli Sillanpää, Emilio Gómez-Lázaro, Richard Scharff, Lennart Söder, Xiaoli Guo Larsén, Gregor Giebel, et al. Wind power forecasting error distributions: An international comparison. In *11th Annual International Workshop on Large-Scale Integration of Wind Power into Power Systems as well as on Transmission Networks for Offshore Wind Power Plants Conference*, 2012.

[JI06] S. Jebaraj and S. Iniyan. A review of energy models. *Renewable and Sustainable Energy Reviews*, 10(4):281 – 311, 2006.

[JL96] George H John and Pat Langley. Static versus dynamic sampling for data mining. In *KDD*, volume 96, pages 367–370, 1996.

[JPK15a] Kristina Jurković, Hrvoje Pandšić, and Igor Kuzle. Review on unit commitment under uncertainty approaches. In *Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2015 38th International Convention on*, pages 1093–1097. IEEE, 2015.

[JPK15b] K. Jurković, H. Pandšić, and I. Kuzle. Review on unit commitment under uncertainty approaches. In *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 1093–1097, May 2015.

[JSCT14] Rishee K. Jain, Kevin M. Smith, Patricia J. Culligan, and John E. Taylor. Forecasting energy consumption of multi-family residential buildings using support vector regression: Investigating the impact of temporal and spatial monitoring granularity on performance accuracy. *Applied Energy*, 123:168 – 178, 2014.

[KRK16] Esmaeil Keyvanshokooh, Sarah M Ryan, and Elnaz Kabir. Hybrid robust and stochastic optimization for closed-loop supply chain network design using accelerated benders decomposition. *European Journal of Operational Research*, 249(1):76–92, 2016.

[KSHdM02] Anton J Kleywegt, Alexander Shapiro, and Tito Homem-de Mello. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12(2):479–502, 2002.

[KW03a] Michal Kaut and Stein W. Wallace. *Evaluation of scenario-generation methods for stochastic programming*. Humboldt-Universität zu Berlin, Mathematisch-Naturwissenschaftliche Fakultät II, Institut für Mathematik, 2003.

[KW03b] Michal Kaut and Stein W. Wallace. *Evaluation of scenario-generation methods for stochastic programming*. Humboldt-Universität zu Berlin, Mathematisch-Naturwissenschaftliche Fakultät II, Institut für Mathematik, 2003.

[KWK94] Peter Kall, Stein W Wallace, and Peter Kall. *Stochastic programming*. Springer, 1994.

[LAGW08] Pu Li, Harvey Arellano-Garcia, and Günter Wozny. Chance constrained programming approach to process optimization under uncertainty. *Computers & chemical engineering*, 32(1-2):25–45, 2008.

[LIDLC+16] Manuel López-Ibáñez, Jérémie Dubois-Lacoste, Leslie Pérez Cáceres, Mauro Birattari, and Thomas Stützle. The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, 3:43–58, 2016.

[LL93a] Gilbert Laporte and François V Louveaux. The integer l-shaped method for stochastic integer programs with complete recourse. *Operations research letters*, 13(3):133–142, 1993.

[LL93b] Gilbert Laporte and François V Louveaux. The integer l-shaped method for stochastic integer programs with complete recourse. *Operations research letters*, 13(3):133–142, 1993.

[LLP12] Chungmok Lee, Kyungsik Lee, and Sungsoo Park. Robust vehicle routing problem with deadlines and travel time/demand uncertainty. *Journal of the Operational Research Society*, 63(9):1294–1306, 2012.

[LMB17] Michele Lombardi, Michela Milano, and Andrea Bartolini. Empirical decision model learning. *Artificial Intelligence*, 244:343–367, 2017.

[LPR09] P Lombardi, M Powalko, and K Rudion. Optimal operation of a virtual power plant. In *Power & Energy Society General Meeting, 2009. PES'09. IEEE*, pages 1–6. IEEE, 2009.

[LTY13] Minlong Lin, Ke Tang, and Xin Yao. Dynamic sampling approach to training neural networks for multiclass imbalance classification. *IEEE Transactions on Neural Networks and Learning Systems*, 24(4):647–660, 2013.

[LXT15] Guodong Liu, Yan Xu, and Kevin Tomsovic. Bidding strategy for microgrid in day-ahead market based on hybrid stochastic/robust optimization. *IEEE Transactions on Smart Grid*, 7(1):227–237, 2015.

[MCM+13] Juans Morales, Antonio Conejo, Henrik Madsen, Pierre Pinson, and Marco Zugno. *Integrating renewables in electricity markets: operational problems*, volume 205. Springer Science & Business Media, 2013.

[MH07] Luc Mercier and Pascal Van Hentenryck. Performance analysis of online anticipatory algorithms for large multistage stochastic integer programs. In *IJCAI*, 2007.

[MNP14] Stefano Manzo, Otto Anker Nielsen, and Carlo Giacomo Prato. *Uncertainty calculation in transport models and forecasts*. PhD thesis, PhD thesis, DTU Copenhagen, 2014.

[MTZ60] Clair Miller, Albert Tucker, and Richard Zemlin. Integer programming formulation of traveling salesman problems. *J. ACM*, 7(4):326–329, October 1960.

[MVH07] Luc Mercier and Pascal Van Hentenryck. Performance analysis of online anticipatory algorithms for large multistage stochastic integer programs. In *IJCAI*, pages 1979–1984, 2007.

[MVH08] Luc Mercier and Pascal Van Hentenryck. Amsaa: A multistep anticipatory algorithm for online stochastic combinatorial optimization. In *International Conference on Integration of Artificial Intelligence (AI) and Operations Research (OR) Techniques in Constraint Programming*, pages 173–187. Springer, 2008.

[Nem] Nemirovskiĭ. *Robust Optimization (Princeton Series in Applied Mathematics)*.

[PBBA+11a] R. Palma-Behnke, C. Benavides, E. Aranda, J. Llanos, and D. Saez. Energy management system for a renewable based microgrid with a demand side management mechanism. In *2011 IEEE Symposium on Computational Intelligence Applications In Smart Grid (CIASG)*, pages 1–8, April 2011.

[PBBA+11b] Rodrigo Palma-Behnke, Carlos Benavides, E Aranda, Jacqueline Llanos, and Doris Sáez. Energy management system for a renewable based microgrid with a demand side management mechanism. In *Computational intelligence applications in smart grid (CIASG), 2011 IEEE symposium on*, pages 1–8. IEEE, 2011.

[PD11] Peter Palensky and Dietmar Dietrich. Demand side management: Demand response, intelligent energy systems, and smart loads. *IEEE transactions on industrial informatics*, 7(3):381–388, 2011.

[PDM12] Andrew B Philpott and Vitor L De Matos. Dynamic sampling algorithms for multi-stage stochastic programs with risk aversion. *European Journal of Operational Research*, 218(2):470–483, 2012.

[Pow16] Warren B. Powell. *A Unified Framework for Optimization Under Uncertainty*, chapter 3, pages 45–83. http://pubsonline.informs.org/doi/pdf/10.1287/educ.2016.0149, 2016.

[Put14] Martin L Puterman. *Markov Decision Processes.: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2014.

[RBG11] F.G. Montoya C. Gil A. Alcayde R. Banos, F. Manzano-Agugliaro and J. Gomez. Optimization methods applied to

renewable and sustainable energy: A review. *Renewable and Sustainable Energy Reviews*, 15(4):1753 – 1766, 2011.

[RM15] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. *arXiv preprint arXiv:1505.05770*, 2015.

[Ros14] Sheldon M Ross. *Introduction to stochastic dynamic programming*. Academic press, 2014.

[RSJ17] S Surender Reddy, Vuddanti Sandeep, and Chan-Mook Jung. Review of stochastic optimization methods for smart grid. *Frontiers in Energy*, pages 1–13, 2017.

[Sah04] Nikolaos V. Sahinidis. Optimization under uncertainty: state-of-the-art and opportunities. *Computers & Chemical Engineering*, 28(6):971 – 983, 2004. FOCAPO 2003 Special issue.

[SE07] Kaplanis S. and Kaplani E. A model to predict expected mean and stochastic hourly global solar radiation i(h;nj) values. *Renewable Energy*, 32(8):1414 – 1425, 2007.

[Sha08] Alexander Shapiro. Stochastic programming approach to optimization under uncertainty. *Mathematical Programming*, 112(1):183–220, Mar 2008.

[Sha13a] Alexander Shapiro. Sample average approximation. In *Encyclopedia of Operations Research and Management Science*, pages 1350–1355. Springer, 2013.

[Sha13b] Alexander Shapiro. Sample average approximation. In *Encyclopedia of Operations Research and Management Science*, pages 1350–1355. Springer, 2013.

[Sil18] Bernard W Silverman. *Density estimation for statistics and data analysis*. Routledge, 2018.

[SP07] Alexander Shapiro and Andy Philpott. A tutorial on stochastic programming. *Manuscript. Available at www2. isye. gatech. edu/ashapiro/publications. html*, 17, 2007.

[TDVWDK13] Duygu Tacs, Nico Dellaert, Tom Van Woensel, and Ton De Kok. Vehicle routing problem with stochastic travel times including soft time windows and service costs. *Computers and Operations Research*, 40(1):214–224, 2013.

[TV02] Paolo Toth and Daniele Vigo. *The vehicle routing problem*. SIAM, 2002.

[VHBV06] Pascal Van Hentenryck, Russell Bent, and Yannis Vergados. Online stochastic reservation systems. In *International Conference on Integration of Artificial Intelligence (AI) and Operations Research (OR) Techniques in Constraint Programming*, pages 212–227. Springer, 2006.

[Wal02] Toby Walsh. Stochastic constraint programming. In *ECAI*, volume 2, pages 111–115, 2002.

[WF03] Stein W. Wallace and Stein-Erik Fleten. Stochastic programming models in energy. In *Stochastic Programming*, volume 10 of *Handbooks in Operations Research and Management Science*, pages 637 – 677. Elsevier, 2003.

[Win04] Wayne Leslie Winston. *Operations research: applications and algorithms*, volume 3. 2004.

[ZG13] C. Zhao and Y. Guan. Unified stochastic and robust unit commitment. *IEEE Transactions on Power Systems*, 28(3):3353–3361, Aug 2013.

[ZL11] Hui Zhang and Pu Li. Chance constrained programming for optimal power flow under uncertainty. *IEEE Transactions on Power Systems*, 26(4):2417–2424, 2011.

[ZWL15] Q. P. Zheng, J. Wang, and A. L. Liu. Stochastic optimization for unit commitment, a review. *IEEE Transactions on Power Systems*, 30(4):1913–1924, July 2015.

[ZZL+13a] Zhe Zhou, Jianyun Zhang, Pei Liu, Zheng Li, Michael C Georgiadis, and Efstratios N Pistikopoulos. A two-stage stochastic programming model for the optimal design of distributed energy systems. *Applied Energy*, 103:135–144, 2013.

[ZZL+13b] Zhe Zhou, Jianyun Zhang, Pei Liu, Zheng Li, Michael C. Georgiadis, and Efstratios N. Pistikopoulos. A two-stage stochastic programming model for the optimal design of distributed energy systems. *Applied Energy*, 103:135 – 144, 2013.